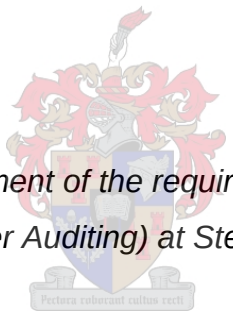
Developing an audit planning framework at a strategic and operational level for implementing continuous auditing and the corresponding continuous auditing procedures for Oracle database management systems

by

Hendrike Olet van Dyk



Thesis presented in partial fulfilment of the requirements for the degree Master of Commerce (Computer Auditing) at Stellenbosch University.

Supervisor: Ms. Riana Goosen

Faculty of Economic and Management Sciences

March 2017

DECLARATION

By submitting this thesis/dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2017

ABSTRACT

Information technology (IT) has become imperative to most modern organisations' strategic and operational activities. It is for this reason that King III clarified the respective responsibilities of risk committees, audit committees and internal audit functions with respect to IT assurance. King III recommends the use of technology to improve audit coverage and audit efficiency, but does not elaborate on this recommendation. In this research study, a modern audit methodology, namely continuous auditing, was explored as a potential solution to address this recommendation made by King III.

Continuous auditing is the ongoing assessment of risks and controls which is enabled by technology. Compared to traditional audit methodologies, continuous auditing is considered a cost-effective method to increase audit efficiency and audit coverage. Despite the stated benefits of this audit methodology, internal auditors are yet to optimise the implementation of continuous auditing in practice.

The primary objective of this research was to develop an audit planning framework for internal auditors to implement continuous auditing to ensure ongoing assurance for automated IT controls. The framework consists of strategic planning steps to develop an annual audit plan and to identify areas where continuous auditing could be implemented. The operational elements of this framework focus only on developing continuous auditing for automated IT controls. The secondary objective was to apply this planning framework to compile continuous audit procedures for database management systems, using Oracle Database as an example.

The degradation of IT controls is often an early-warning indicator of fraud and error. The implementation of this modern audit methodology for database management systems enables internal auditors to report on control deficiencies within a shorter timeframe to provide real-time assurance. Considering that the most valuable information assets are retained in databases and in view of the increase in data breach incidents involving high-profile organisations, the implementation of continuous controls auditing should be a high priority for internal audit functions.

OPSOMMING

Inligtingstegnologie (IT) het die middelpunt van die meeste hedendaagse organisasies se strategiese en operasionele aktiwiteite geword. Om hierdie rede het King III die onderskeie verantwoordelikhede van risikokomitees, ouditkomitees en interne ouditfunksies met betrekking tot gerusstelling vir IT-stelsels uiteengesit. King III beveel aan dat tegnologie gebruik moet word om die effektiwiteit en dekking van oudits te verbeter, maar brei nie uit op hierdie aanbeveling nie. In hierdie studie word 'n moderne ouditmetode, naamlik deurlopende ouditering, ondersoek as 'n potensiële oplossing vir hierdie aanbeveling van King III.

Deurlopende ouditering is die voortdurende assessering van risiko's en kontroles wat deur tegnologie moontlik gemaak word. In vergelyking met tradisionele ouditmetodes, word deurlopende ouditering beskou as 'n koste-effektiewe metode om oudit-effektiwiteit en dekking te verhoog. Ten spyte van die genoemde voordele van hierdie ouditmetode, het interne ouditeure nog nie deurlopende ouditering optimaal in die praktyk geïmplementeer nie.

Die primêre doel van hierdie navorsing was om 'n oudit-beplanningsraamwerk vir interne ouditeure te ontwikkel om deurlopende ouditering vir IT-stelsels te implementeer. Die raamwerk bestaan eerstens uit strategiese beplanningstappe om 'n oorhoofse ouditplan te ontwikkel om sodoende areas te identifiseer waar deurlopende ouditering gebruik kan word. Daarna fokus die operasionele elemente van die raamwerk slegs op die implementering van deurlopende ouditering vir outomatiese IT-kontroles. Die sekondêre doel van hierdie navorsing was om hierdie beplanningsraamwerk te gebruik om deurlopende ouditprosedures vir databasis-bestuurstelsels saam te stel, met Oracle Database as voorbeeld.

Die agteruitgang van IT-kontroles is dikwels 'n vroeë aanduider van bedrog en foute. Die implementering van hierdie moderne ouditmetode vir die ouditering van databasis-bestuurstelsels stel interne ouditeure binne 'n korter tyd in staat om verslag te lewer oor kontrolegebreke, om sodoende deurlopende gerusstelling te bied. Aangesien die waardevolste inligtingsbates in databasisse gestoor word, en in die lig van die verhoging in insidente van datadiefstal by hoëprofiel-organisasies, behoort die implementering van deurlopende ouditering 'n hoë prioriteit vir interne ouditfunksies te wees.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION 1

1.1 Introduction and background 1

1.2 Research problem and motivation 2

1.3 Research objective and scope 3

1.4 Research methodology 4

1.5 Organisation of the research 6

CHAPTER 2. HISTORICAL RESEARCH 8

2.1 Introduction 8

2.2 Industry studies 8

2.3 Academic research 10

2.4 Professional accounting and auditing associations 12

2.5 Technical literature 13

2.6 Conclusion 14

CHAPTER 3. LITERATURE REVIEW: DEFINITION AND SCOPE OF CONTINUOUS AUDITING 15

3.1 Introduction 15

3.2 Continuous auditing definition 16

3.3 The value of continuous auditing 17

3.4 External versus internal audit 17

3.5 Comparison of traditional and continuous auditing methodologies 18

3.6 The relationship between data analysis and continuous auditing 19

3.7 The elements of continuous auditing 21

3.7.1 Continuous data auditing 21

3.7.2 Continuous control monitoring 22

3.7.3 Continuous risk monitoring 22

3.7.4 Continuous compliance monitoring 23

3.8 Continuous monitoring 23

3.9 Continuous assurance 25
3.10 Conclusion 25
CHAPTER 4. FINDINGS: AUDIT PLANNING FRAMEWORK AT A STRATEGIC AND OPERATIONAL LEVEL FOR IMPLEMENTING CONTINUOUS AUDITING 28
4.1 Introduction 28
4.2 Level I: Develop the overall audit strategy and plan 29
4.2.1 Develop the audit universe 30
4.2.2 Perform high-level risk assessment 31
4.2.3 Develop high-level annual audit plan 31
4.2.4 Perform maturity assessment for continuous auditing activities 31
4.3 Level II: Develop a continuous audit implementation plan for selected business processes 33
4.4 Level III: Perform an application risk assessment using access paths 35
4.5 Level IV: Develop continuous audit procedures for individual access path components 37
4.5.1 Determine the product's lifecycle phase 38
4.5.2 Define risk and control indicators (baseline standards) 39
4.5.3 Audit software/tool selection 42
4.5.3.1 Generalised audit software 43
4.5.3.2 Generic vulnerability assessment tools 44
4.5.3.3 Specific vulnerability assessment tools 45
4.5.3.4 Password strength and hacking tools 45
4.5.4 Report and manage results 45
4.6 Conclusion 46
CHAPTER 5. FINDINGS: DEVELOPING CONTINUOUS AUDITING PROCEDURES FOR ORACLE DATABASE MANAGEMENT SYSTEMS 49
5.1 Introduction 49
5.2 Configurable controls for database management systems 50
5.3 Database vulnerabilities: Product version and patch management 53
5.3.1 Product version 53

5.3.2 Patch management 54
5.4 Account and password management 58
5.4.1 User account management 58
5.4.2 Default accounts and passwords 60
5.4.3 Password management capabilities 62
5.5 Database permissions management 67
5.5.1 Database permissions – background 67
5.5.2 Review database privileges granted to end-users 68
5.5.3 Implicit database permissions 70
5.5.4 Row-level access to table data 71
5.5.5 *PUBLIC* permissions 72
5.6 Database auditing and monitoring 77
5.6.1 Types of database auditing 77
5.6.2 Enabling database auditing 81
5.6.3 Protecting the audit trail 83
5.6.4 Stored procedures database triggers 85
5.7 Conclusion 90
CHAPTER 6. CONCLUSION 91
APPENDIX 1 – ACCESS PATH COMPONENTS 93
APPENDIX 2 – DATABASE AUDITING PARAMETERS 94
REFERENCES 96

LIST OF TABLES

Table 3.1 Comparison of traditional and continuous auditing methodologies 19
Table 4.1 Level I: Audit methodology-based maturity assessment model 32
Table 4.2 Description of product lifecycle phases 38
Table 4.3 Level IV: Example of a continuous baseline standard comparison 40
Table 5.1 Continuous audit procedures: Database vulnerabilities 57
Table 5.2 Oracle DBA tables for user access review 59
Table 5.3 Examples of Oracle default accounts 60
Table 5.4 Description of recommended Oracle 12c password parameters 63
Table 5.5 Continuous audit procedures: User account and password management 65
Table 5.6 Continuous audit procedures: Permissions management 74
Table 5.7 Continuous audit procedures: Database monitoring and auditing 86

LIST OF FIGURES

Figure 3.1 The elements of continuous auditing 21
Figure 3.2 The relationship between continuous auditing and continuous monitoring 24
Figure 3.3 Continuous assurance 25
Figure 3.4 The evolution from traditional auditing to continuous auditing 26
Figure 4.1 Continuous auditing planning levels 28
Figure 4.2 Level I: Develop the overall audit strategy and plan 30
Figure 4.3 Level II: Evaluate business processes for continuous auditing 34
Figure 4.4 Level III: Multi-tier model to identify IT architectural components 35
Figure 4.5 Level III: Simplified example of an IT access path 36
Figure 4.6 Level IV: Develop continuous auditing procedures for selected access path component 37
Figure 4.7 Level IV: Identify the lifecycle phase for access path components 39
Figure 4.8 Level IV: Continuous auditing using a baseline standard 41
Figure 4.9 Level IV: Selecting audit software/tools 43
Figure 4.10 Planning framework for developing continuous auditing procedures 48
Figure 5.1 Control categories for database management systems 50
Figure 5.2 Role versus privilege assignment for user accounts 68
Figure 5.3 Types of database auditing 78
Figure 5.4 Oracle unified auditing – mixed mode 80

CHAPTER 1. INTRODUCTION

1.1 Introduction and background

The King reports on governance principles have formed the basis for good corporate governance practices in South African organisations for the past two decades (Goosen, 2012). However, information technology (IT) governance was only addressed for the first time in the third King report (King III) (Institute of Directors (IODSA), 2009). The IT governance chapter of King III covers the salient aspects of IT governance-related matters and also states the responsibilities of the risk committee, audit committee and internal audit function with respect to the IT assurance function (IODSA, 2009).

King III states that the risk and audit committees should assist the board of directors in carrying out its IT responsibilities (IODSA, 2009). Risk committees are advised to obtain appropriate assurance that IT risks are appropriately governed and that sufficient controls are in place to address IT risks (IODSA, 2009). One of the primary responsibilities of the internal audit function is to report to the organisation's board of directors on IT risk assurance matters (IODSA, 2009). In particular, King III (principle 5.7 paragraph 48) recommends that the audit committee should consider using technology and related techniques to improve audit coverage and audit efficiency (IODSA, 2009). In this research study, a modern audit methodology, namely continuous auditing, was explored as a potential solution to address this recommendation of King III.

The Institute of Internal Auditors (IIA) defines continuous auditing as ongoing risk and control assessments which are enabled by technology (IIA, 2015). Compared to traditional audit methodologies, continuous auditing is considered a cost-effective method to increase audit efficiency and audit coverage (Whitehouse, 2012). Traditional auditing techniques are often of manual nature and the frequency of audits is limited to annual or bi-annual reviews (IIA, 2015). As a result, material errors, omissions or fraud incidents may not be detected until the annual audit is conducted (Chan & Vasarhelyi, 2011). In comparison, continuous auditing is an audit methodology that enables auditors to gather audit evidence through the use of a computer on a continuous basis, which may detect irregular instances in a timely manner (ISACA, 2016).

Although data analysis was mentioned in auditing standards as early as 1978 (Soileau, Soileau & Sumners, 2015), industry studies conducted by Protiviti (2015a), the Corporate Executive Board (CEB, 2015) and PricewaterhouseCoopers (PwC, 2015) concluded that internal auditors have not yet leveraged the benefits of technology-enabled continuous auditing techniques in their audit procedures.

1.2 Research problem and motivation

Although King III recommends the use of technology to improve audit coverage and efficiency, it does not elaborate on this recommendation (IODSA, 2009). Therefore, this research explores continuous audit methodologies as an alternative to traditional audit techniques, focusing on the internal audit function's role to provide assurance on IT risks, among other recommendations.

Academics and internal audit practitioners agree that continuous auditing can increase audit productivity and efficiency (Chan & Vasarhelyi, 2011). It also increases audit coverage and effectiveness, resulting in increased confidence in the audit procedures performed (Soileau *et al*., 2015). However, despite the stated benefits, continuous auditing remains mostly underutilised by internal audit functions and the implementation of this methodology remains on the agenda for internal auditors globally (Deloitte, 2016; PwC, 2015; Protiviti, 2015a; CEB, 2015). In particular, PwC (2013) reported that internal audit functions lack the required skill and capacity to utilise technology to perform a more effective audit by utilising continuous auditing techniques in an efficient manner, in both their audit planning procedures as well as audit fieldwork.

Continuous auditing is therefore considered an emerging research area (Chiu, Liu & Vasarhelyi, 2014), with a low adoption rate in practice (PwC, 2015). Industry studies confirmed that the implementation and improvement of continuous auditing initiatives continue to be a focus area for internal audit practitioners (Deloitte, 2016; PwC, 2015). Current initiatives are mostly immature in nature and include only limited transactional data analysis (Protiviti, 2015a).

Continuous auditing methodologies are also applicable to the automated IT system controls. The degradation of IT controls often occurs in advance of the symptomatic errors in transactional data and the ongoing assessment of controls enables internal auditors to

provide management with an early warning of control deficiencies and violations (IIA, 2015). In this manner, internal auditors are enabled to provide assurance on IT risks relating to key information assets such as databases. Considering the absence of readily available continuous auditing procedures for automated IT controls, guidance is needed to assist internal audit practitioners to implement continuous audit methodologies practically.

Since organisations retain valuable data in databases, database management systems are often the target of security breaches (Davis, Schiller & Wheeler, 2011). Perimeter security protection such as firewalls is no longer considered sufficient to protect data assets and the focus has shifted to protecting data at the source, i.e. databases (Davis *et al*., 2011). As such, the risks related to the validity and integrity of data should be of concern to audit committees and internal audit functions (IODSA, 2009).

However, limited literature is available to guide internal audit functions to implement this modern audit methodology as an alternative method to provide assurance for automated IT controls, specifically relating database management systems. Continuous audit procedures are therefore developed for database management systems in this study to address this gap.

1.3 Research objective and scope

The primary objective of this research was to develop an audit planning framework for internal auditors to provide assurance through the implementation of continuous audit methodologies. This framework provides guidance for audit planning at a strategic and operational level. The strategic level entails processes to develop an overall audit plan and steps to identify areas suited for implementing continuous auditing. At an operational level, one of four elements of continuous auditing, namely continuous control monitoring, is further discussed in detail to describe the planning steps to implement ongoing control assessments for automated IT controls. The remaining three elements of continuous auditing as defined by Bumgarner and Vasarhelyi (2015), namely continuous data auditing, risk monitoring and compliance monitoring, are excluded from this study. These elements consist mainly of transactional data analysis, compared to the continuous assessment of automated IT controls **(continuous controls monitoring)** (Bumgarner & Vasarhelyi, 2015), which is the focus of this study.

The secondary objective of this research was to apply the above-mentioned planning framework to compile a list of continuous audit procedures specifically for Oracle database management systems. Oracle Database was chosen since it was identified in Gartner's 2015 magic quadrant report as one of the two leaders for operational database management systems (Feinberg, Adrian, Heudecker, Ronthal & Palanca, 2015). Only the controls and procedures relating to the validity, integrity and confidentiality of data are included in this research, considering the commercial value of the data retained in databases (refer to paragraph 1.2). The controls that ensure system availability are excluded.

This study was limited to the utilisation of generalised audit software which does not operate on a truly continuous basis. Instead, generalised audit software represents batch programs that are activated periodically (e.g. daily, weekly or monthly) according to the audit objectives and risk assessment (Byrnes, Al-Awahdi, Gullvist, Brown-Liburd, Teeter, Warren & Vasarhelyi, 2015b). Alternative approaches that may in future provide true continuous auditing solutions include the following:

- Embedded audit modules (EAM) that involve the installation of coded segments within the host system to provide an integrate test facility;
- Monitoring and controls layer (MCL) architecture, which is a middleware solution that extracts data from disparate systems for further analysis;
- An audit data warehouse model that entails extracting and transforming data in real time and is made available in audit-specific data marts (Byrnes *et al.,* 2015b).

However, due to the various concerns noted with each of the above approaches, Byrnes *et al.* (2015b) observed that the above-mentioned alternatives still remain as academic topics only. Concerns include the high implementation cost, potential impairment of auditor independence and challenges in securing the data and logs from manipulation by IT staff (Byrnes *et al*., 2015b). Therefore, the audit planning framework and audit procedures in this study was developed considering the capabilities of generalised audit software.

1.4 Research methodology

The research problem was addressed by conducting a non-empirical study of existing literature from accredited academic articles in international journals, electronic sources,

White Papers, theses and academic text books. Where applicable, auditing standards published by the IIA and ISACA (previously known as the Information Systems Audit and Control Association) were also consulted. Technical resources included the best practice standards published by software companies such as Oracle and the security benchmarks published by the Centre for Internet Security (CIS). The following aspects were researched:

- The definition and scope of continuous auditing and related topics;
- Historical literature that demonstrates the importance of continuous auditing and the perceived underutilisation for this modern audit methodology;
- Implementation guidance to change audit procedures from traditional auditing to continuous auditing techniques;
- Auditing procedures relevant to database management systems, including configuration controls which can be audited using computer-assisted audit tools and techniques (CAATTs).

Based on the literature review, it was possible to develop an audit planning framework in order to implement continuous auditing processes, which was then applied to develop continuous audit procedures for the Oracle database management system. A three-step approach was followed:

Step 1: Continuous auditing was defined and distinguished from traditional auditing methodologies in Chapter 3.

Step 2: A framework was developed to provide guidance to internal auditors when planning the implementation of continuous auditing techniques. The framework entails four levels of detail, as discussed in Chapter 4.

Level I: A continuous auditing implementation strategy is developed which is embedded in the strategic audit plan and the resulting annual audit plan.

Level II: The implementation strategy is further refined by performing a risk and control assessment for selected business processes which forms the foundation for developing detailed continuous audit procedures.

Level III: At an operational level, the different IT access paths of a particular business process are analysed to ensure that all the underlying IT architectural

components are identified. A risk and control assessment is conducted for each component.

Level IV: Detailed continuous auditing procedures are developed for the particular access path component under review. The continuous audit procedures are determined by considering the lifecycle phase of the product's development and the risks and controls relating to the process and component under review. Baseline standards are developed for key controls to be tested continuously, using automated tools. The specific tools to automate the process, such as generalised audit software, are selected at this stage.

Step 3: Using the framework developed in step 2, practical continuous auditing procedures were then developed for one access path component, namely database management systems, as discussed in Chapter 5. Oracle Database was used as an example. This was done by firstly describing the risks and controls for each control area, as well as the relating traditional and continuous audit procedures for each identified control area. The continuous auditing procedures were then tabled for each lifecycle phase, where applicable. These continuous auditing procedures were developed considering the capabilities of generalised audit software.

1.5 Organisation of the research

The thesis consists of the following chapters**:**

Chapter 1: Introduction. Following an introduction, the research problem and motivation and research methodology are discussed.

Chapter 2: Historical research. A historical literature review demonstrates the emerging nature of continuous auditing as a research area for various interested stakeholders. In this chapter, the development of continuous auditing as an academic topic is summarised, together with the adoption of this audit methodology by internal audit functions. Considering the low adoption rate observed in practice, the guidance on this topic offered by accounting and auditing associations is also evaluated together with available technical literature developed by inter alia software companies. It is concluded in this chapter that detailed guidance have not yet been documented for the implementation of continuous

audit procedures for automated IT controls, specifically for database management systems.

Chapter 3: Literature review: Definition and scope of continuous auditing. A literature review clarifies the definition and scope of continuous auditing, in comparison with related terminology such as data analysis and continuous monitoring. The evolution of data analytics to continuous auditing and, ultimately, to continuous assurance is demonstrated in this chapter.

Chapter 4: Findings: Audit planning framework at a strategic and operational level for implementing continuous auditing. A generic audit planning framework is developed at a strategic and operational level to guide internal audit practitioners when implementing the continuous auditing methodology. The focus is on one element of continuous auditing, namely continuous controls monitoring of automated IT controls. The planning steps are summarised in the framework consisting of four levels.

Chapter 5: Findings: Continuous auditing procedures for Oracle database management systems. A literature review is performed to identify the risks and controls relevant to database management systems, using Oracle Database as an example. A practical implementation guide is developed listing the continuous audit procedures for each control area, considering the different phases of the product's lifecycle, relating specifically to Oracle Database.

Chapter 6: Conclusion. An overview of the research, highlighting the outcomes of the research, is provided in this chapter. Areas relating to this topic that remain available for future research are also identified.

CHAPTER 2. HISTORICAL RESEARCH

2.1 Introduction

Continuous auditing is considered an emerging research area (Chiu *et al*., 2014) for various stakeholders. In particular, the majority of academic contributions have so far focused on the consequences and benefits of continuous auditing as well as on certain technical aspects, such as the architectural design aspects, of implementing continuous auditing technologies (Chiu *et al.*, 2014). In recognition of the benefits of this audit methodology, accounting and auditing associations have also invested in developing guidance on continuous auditing (AICPA, 2015; IIA, 2015; ISACA, 2010). This guidance is however introductory in nature and offers mainly strategic implementation guidance, without detailing continuous auditing procedures at an operational level for any particular IT architecture component. Despite the repeated optimism demonstrated by internal audit practitioner surveys, it appears as if the implementation of this audit methodology has however advanced very slowly in practice (Gonzalez, Sharma & Galletta, 2012).

2.2 Industry studies

The emerging nature and low adoption rate of continuous auditing in practice was confirmed by industry studies published by audit and consulting firms in 2015 and 2016, as illustrated below. These studies researched internal audit functions worldwide and are conducted periodically to identify focus areas and opportunities for enhancement of audit capabilities.

- Continuous auditing and CAATTs, combined with data mining and data analysis tools, remained on the agenda for internal audit leaders since 2013, according to Protiviti's annual *2015 Internal Audit Capabilities and Needs Survey* with more than 800 correspondents (Protiviti, 2015a). A follow-up survey, focusing on data analytics and continuous auditing, found that internal audit functions consider data analytics as a high priority and that there are significant opportunities to expand continuous auditing initiatives (Protiviti, 2015b). Both studies provided recommendations for internal audit functions to improve their analytical capabilities.

- Similarly, the CEB *2015 Audit Department Challenges and Priorities* survey, involving more than 100 internal audit functions, confirmed that the implementation and improvement of data analytics are the most significant priorities for internal audit functions. The advancement of data analytics capabilities were noted as either a high or very high priority for 2015 by 52% of respondents, while 35% rated this as a moderate priority (CEB, 2015).
- PwC (2013) reported that internal audit functions lacked the necessary skill and capacity to utilise technology to perform a more effective audit. Less than a third of the respondents indicated that they were using data analytics on a regular basis (Le Roux & Wallis, 2014; PwC, 2013). Although limited improvement was noted in 2015, data analysis was identified as one of four focus areas for internal audit functions (PwC, 2015). PwC's *2015 State of the Internal Audit Profession s*tudy, involving more than 1 300 chief audit executives, revealed that most internal audit functions are still considering how data analytics can be leveraged more efficiently and effectively. Most functions are experimenting with expanding the use of data analysis (PwC, 2015). While 82% of chief audit executives indicated that data analytics are used in specific audits, 48% use analytics for scoping decisions and 43% leverage data as part of risk assessments (PwC, 2015). It can be concluded that data analysis is not yet embedded throughout all audit processes, including annual planning, engagement planning and audit field work, while continuous auditing is still in an immature state in practice (PwC, 2015).
- Deloitte (2016) reported similar findings in their global survey involving approximately 1 200 chief audit executives. It was found that 86% of respondents use data analytics, but only 24% rated its usage at an intermediate level and 7% at an advanced level. The primary area of usage was audit field work (66%), followed by engagement planning (36%) and annual planning (32%) (Deloitte, 2016).
- AuditNet's 2012 survey report on data analysis software concluded that internal auditors were using data analysis software mainly on an ad hoc basis (AuditNet, 2012). A follow-up survey conducted in 2015 indicated that 60% of the respondents have purchased analytical software. However, only 24% of the respondents indicated that they always use data analysis to develop the annual audit plan, while 68% included data analysis in audit fieldwork, only on an ad hoc basis (AuditNet, 2015).

- The strategic importance of both data analysis and continuous auditing was confirmed in the IIA's 2015 Common Body of Knowledge (CBOK) survey involving 14 500 internal audit practitioners (IIA Research Foundation, 2015). Compared to CBOK 2006, CBOK 2015 shows a 14% increase in the use of technology tools, particularly in the use of data mining (IIA Research Foundation, 2015). Currently, 53% of respondents are moderately or extensively involved in data mining (Cangemi, 2016). However, continuous auditing is one of the least used technology techniques indicated in the 2015 survey and is used extensively by only 14% of respondents, with a 7% increase observed from 2006 (Cangemi, 2016).

It is evident from the above industry studies that internal audit practitioners have not yet optimised the use of data analytics, which is a precursor for continuous auditing. Current initiatives are mostly limited to transactional analytics and have not necessarily evolved to the continuous assessment of automated controls. The low adoption rate of this modern audit methodology observed by industry studies, was also confirmed in academic research.

2.3 Academic research

The concept of continuous auditing first transpired in academic research in the late 1980s and early 1990s. Vasarhelyi (1983) is considered the first academic to commence with researching opportunities to implement technology to aid the execution of audit tasks. Initial research included examining the evolution of automated audit processes (Chiu *et al*., 2014). Computerised audit implementations only reflected the computerisation of manual methods rather than the re-engineering of associated audit processes (Vasarhelyi, 1984). Since the 1980s, more researchers demonstrated the potential of "closer to the event" assurance processes, namely continuous auditing (Groomer & Murthy, 1989; Vasarhelyi & Halper, 1991). Authors have questioned the timeliness, efficiency and appropriateness of traditional audit procedures, where financial statements are audited months after the occurrence of the actual business activities (Bumgarner & Vasarhelyi, 2015)

An increase in academic interest in continuous auditing was noted from 2001 (Chiu *et al.*, 2014). Academic studies conducted between 2000 and 2014 further emphasised the need for continuous auditing by evaluating the methodology, costs, benefits and enabling technologies (Chiu *et al.*, 2014). In this period, research extended to case studies which

analyse the utilisation of this audit methodology in practice, including analyses of the enabling technologies (Chiu *et al.*, 2014). The main focus areas were financial statement and transactional analysis (Byrnes, Ames, Vasarhelyi, Pawlicki & McQuilken, 2015a; Alles, Kogan & Vasarhelyi, 2011).

Academics also commenced with developing frameworks to assist audit practitioners in transforming the traditional manual audit processes to an automated process and potentially real-time reporting (Flowerday, Blundell & Von Solms, 2006). Continuous monitoring and continuous assurance studies were also conducted in this period (Alles, Brennan, Kogan, & Vasarhelyi, 2006).

The idea of continuous auditing was initially conceptualised as a transaction monitoring and trend analysis function, which could be enhanced with an exception reporting facility (Alles *et al*., 2006). The focus was on the analysis of transactions underlying the annual financial statements, with little mention of the automation of the audit procedures for automated IT controls (Bumgarner & Vasarhelyi, 2015). The continuous audit concept was however expanded to also provide assurance over the adequacy of internal controls (including IT configuration controls) as a response to the Sarbanes-Oxley Act of 2002 (Bumgarner & Vasarhelyi, 2015).

Later studies focused on continuous auditing of automated IT controls. Alles *et al*. (2006) studied a methodology where auditors are alerted of any changes to configuration settings of an enterprise resource planning (ERP) system which is compared to a baseline standard of configuration settings. The original work of Alles *et al*. (2006) was subsequently extended to a wider set of controls and parameters (Teeter, 2014). Audit automation, remote auditing and continuous auditing were joined in a framework to assist auditors in identifying opportunities for audit innovation (Teeter, 2014).

Chiu *et al*. (2014) concluded that continuous auditing can be considered an emerging research area, with architectural issues, such as technical implementation challenges relating to continuous auditing, being the most prevalent subject matter, followed by studies focusing on the consequences of implementing the continuous auditing techniques. Despite the increased academic interest in continuous auditing noted since 2000 (Chiu *et al*., 2014), organisations are not yet reaping the benefits of this advanced audit methodology (Byrnes *et al*., 2015a). It is therefore not surprising that accounting and

auditing associations continue to invest in continuous auditing guidance and studies, still attempting to find an effective and practical methodology for implementing such procedures.

2.4 Professional accounting and auditing associations

The first guidance on continuous auditing by accounting and auditing associations was jointly published in 1999 by the Canadian Institute of Chartered Accountants (CICA) and the American Institute of Certified Public Accountants (AICPA) (CICA & AICPA, 1999). This publication was superseded in 2015 by a compendium of academic essays which provide an overview of continuous audit theory and practice (AICPA, 2015).

Following CICA & AICPA (1999), the IIA Research Foundation published a research report in 2003 that explained the concept and benefits of continuous auditing and also provided practical implementation guidance (Warren & Parker, 2003). The IIA research report was complemented in 2005 by the IIA's *GTAG 3 Continuous Auditing: Implications for Assurance* (IIA, 2005). The second edition of *GTAG 3* was published in 2015. This guidance consists of foundational and optimised continuous auditing assurance frameworks and includes updated practical applications for continuous auditing (IIA, 2015). The guidance focuses on planning steps for implementing continuous auditing techniques and includes high-level planning steps for both continuous transactional and controls auditing. Although IIA (2015) addresses strategic and operational planning steps, the guidance relating to continuous control monitoring is at an introductory level. In particular, at an operational level, implementation guidance does not extend to any particular IT architecture component.

ISACA also published guidance on continuous auditing in 2010, titled *G42 IT Audit and Assurance Guidelines: Continuous Assurance* (ISACA, 2010). These guidelines are based on the IIA's *GTAG 3* and are therefore also limited to high-level implementation guidance only. The guidance does not extend to continuous auditing procedures for automated IT controls, but is limited to generic planning steps for transactional auditing only (ISACA, 2010).

Similar to AICPA (2015), the Australian Institute of Chartered Accountants published a White Paper in 2010, which defines continuous auditing and provides limited implementation guidance and introductory examples (Vasarhelyi, Alles & Williams, 2010).

Considering the updated guidance published in 2015 by both the IIA and AICPA, it is evident that continuous auditing remains an emerging and relevant topic for professional accounting and auditing associations. However, the above-mentioned documents contain only high-level implementation guidance which is mainly strategic in nature. On an operational level, the guidance focuses on transactional data analysis and do not extend to detailed guidance for continuous control monitoring of automated IT controls. In particular, the available guidance does not extend to any specific IT architectural component, such as database management systems, as is the objective of this study.

2.5 Technical literature

The abovementioned literature on continuous auditing does not include detailed guidance or auditing procedures for any particular IT architecture component, such as database management systems. There is a variety of technical literature covering the security and configuration of specific software installations, such as software-specific security benchmarks (CIS, 2015), security handbooks (Wright, 2014) and implementation guidance by software vendors (Huey, 2016) which focuses on Oracle Database only. However, these publications focus on the system configuration to be applied by IT management and are not intended to serve as practical continuous auditing procedures.

Furthermore, audit-specific literature focusing on database management systems is limited. Most notable is ISACA (2009) that details security and audit guidance specific to Oracle Database. However, the audit procedures documented by ISACA (2009) are limited to traditional audit procedures for older versions of Oracle Database. To address the perceived underutilisation of automated audit procedures for database management systems, Cooke (2014) proposed that database management systems could be audited using computer assisted audit tools and techniques (CAATTs). The concept was demonstrated for limited configuration settings (mainly user account management) for Oracle Database (Cooke, 2014). This study was followed by a similar high-level article focusing on SQL Server (Cooke, 2015). The audit methodology proposed by Cooke

(2014) was utilised in this study to develop detailed continuous audit procedures for database management systems. In particular, the work of Cooke (2014) is extended in this study to include a broader set of controls for Oracle Database and the related audit procedures that can be repeated continuously.

2.6 Conclusion

It can be concluded from the above studies that, although internal audit practitioners recognise the value of data analysis and continuous auditing, the implementation of this methodology remains low in practice. Continuous auditing and its precursor, data analytics, have remained emerging topics for internal auditors (PwC, 2015).

The underutilisation of this modern auditing methodology in practice (PwC, 2015) may be attributed to the lacking guidance for inter alia the continuous control monitoring of automated IT controls. Although standards and guides developed by professional associations address strategic and operational planning steps to implement continuous auditing, the guidance focuses on continuous auditing using transactional data, while the guidance relating to the continuous assessment of automated controls is at a strategic level.

Furthermore, academic field studies of this methodology focused on continuous data (transactional) auditing, with limited inclusion of continuous auditing procedures relating to automated IT controls. In particular, literature in this area is limited to specific software applications only, mostly related to ERP systems. Limited audit-specific literature relating to Oracle database management systems was found.

Therefore, a detailed audit planning framework was developed in this study to guide the implementation of continuous auditing procedures. The framework includes planning steps at both a strategic and operational level. At an operational level, only those planning step relevant to the continuous assessment of automated IT controls are included. This planning framework is then applied by developing detailed continuous auditing procedures for Oracle database management systems.

CHAPTER 3. LITERATURE REVIEW: DEFINITION AND SCOPE OF CONTINUOUS AUDITING

3.1 Introduction

King III recommends that audit committees should consider the use of technology to improve audit coverage and efficiency (IODSA, 2009). Continuous auditing is considered a cost-effective method to increase such audit coverage and efficiency requirements and is noted as an alternative methodology to traditional audit methodologies (IIA, 2015).

Continuous auditing is attracting increased attention in the internal auditing environment, as discussed in Chapter 2. Although many benefits, including improved efficiencies, have been noted in studies since 1983, internal auditors globally have not yet fully leveraged the benefits of this alternative audit methodology (Byrnes *et al*., 2015a). While continuous auditing is utilised mostly for analysing transactional data, this methodology could be also be used to provide assurance relating to automated IT controls relating to IT architecture components such as operating systems, databases and software applications (IIA, 2015).

The terms **continuous auditing, continuous monitoring** and **continuous assurance are** however often incorrectly used interchangeably:

- **Continuous auditing** refers to the ongoing assessment of risks and controls by internal auditors, which is achieved through automated audit processes (refer to paragraph 3.2) (IIA, 2015).
- **Continuous monitoring** includes management's processes which assess the adequacy of controls and includes those processes that ensure policies are operating effectively. Continuous monitoring is performed by financial, operational and IT management (refer to paragraph 3.8) (IIA, 2015).
- **Continuous assurance** is the result of harmonised continuous auditing techniques and continuous monitoring processes, which is mainly achieved through automation of procedures (refer to paragraph 3.9) (Roth, 2012).

This study focuses on the continuous auditing procedures that could be implemented by internal audit functions. Although procedures may be similar in nature to the continuous monitoring activities conducted by management, internal audit's assurance activities should be conducted independently from management to provide independent assurance to the audit committee (IIA, 2015).

3.2 Continuous auditing definition

The IIA (2015) defines continuous auditing as ongoing risk and control assessments which are enabled by technology. Similarly, ISACA (2016) describes continuous auditing as an approach which enables auditors to monitor system reliability and gather selective audit evidence through the use of a computer on a continuous basis.

Continuous auditing is designed to enable internal auditors to report audit results in a shorter timeframe compared to the traditional retrospective audit approach (IIA, 2015). Continuous audit procedures are dependent on defined processes and enabling technologies (IIA, 2015; Roth, 2012) and could entail any method used by auditors to perform audit-related activities on a continuous basis, ranging from continuous controls assessment to continuous risk assessments (IIA, 2011).

Although continuous auditing could potentially be conducted in real time, the frequency of analysis is determined by the level of risk, the business cycle and the extent and frequency of management's monitoring controls (IIA, 2015). The frequency of transaction exception reporting may coincide with the financial reporting cycle, such as on a monthly or annual basis (IIA, 2015).

Continuous auditing is not limited to transactional analysis only, but may also extend to IT systems, including automated controls and operational IT processes (IIA, 2015). Also, at an operational level, security event monitoring may be conducted in real time for analysis and follow-up as these events occur (Hargenrader, 2015). Since changes to automated or configured controls are typically infrequent, continuous auditing procedures may rather be synchronised with the routine software release and upgrade cycles managed by the organisation's IT department (IIA, 2015).

To enable real-time auditing, technology plays a key role in automating the continuous audit process. These tools are used for the identification of exceptions, trend analysis, detailed transaction analysis, comparisons against thresholds, testing of controls and the comparison of a process or system over time (IIA, 2011).

The four elements of continuous **data auditing, control monitoring, risk monitoring** and **compliance monitoring** are discussed further in paragraph 3.7.

3.3 The value of continuous auditing

Academics and internal audit practitioners have identified a range of benefits originating from continuous auditing, as discussed below.

- Continuous auditing enables auditors to report on a subject matter within a shorter timeframe, potentially in real time or instantaneously (Soileau *et al.*, 2015; ISACA Standards Board, 2002). This could result in more timely (or real-time) risk assurance processes (Chan & Vasarhelyi, 2011). Auditors can therefore actively detect and investigate exceptions as they occur, compared to traditional (annual) auditing processes, which typically detect exceptions long after the actual occurrence thereof (Chan & Vasarhelyi, 2011).
- In addition to transactional analysis, continuous auditing can also be deployed to detect control weaknesses relating to IT systems, thereby enabling the timely remediation by management (IIA, 2015).
- Data analysis technology has enabled auditors to improve the efficiency of audits through the automation of processes (Roth, 2012). Audit functions have also been able to broaden the scope of assurance activities through the automation of analytical procedures, without noting an associated increase in the number of audit staff (Roth, 2012). It has also enabled remote auditing of distributed processes, thereby reducing the travelling costs to remote locations (Teeter, 2014).
- Audit coverage and effectiveness are increased since continuous auditing typically covers the entire transaction population using data analysis (IIA, 2015).
- Data analysis technologies enable auditors to access data independently as they are no longer reliant on the organisation's personnel to extract data. This reduces the opportunity for data manipulation and increases the confidence in the accuracy and completeness of the data being analysed (IIA, 2011).

These benefits have been realised mostly by internal auditors, as discussed in paragraph 3.4 below.

3.4 External versus internal audit

Although international accounting and auditing professional bodies have published guidance on continuous auditing, this methodology is primarily used by internal auditors

(Gonzalez *et al*., 2012). The original development of continuous auditing was aimed at replacing the annual external audit processes. However, external audit firms primarily do not use continuous audit techniques, but rather consult with internal audit functions on this matter (Bumgarner & Vasarhelyi, 2015).

The most prevalent consideration for external auditors is the high implementation cost, compared to the lengthy return period and the short-term nature of external audit engagements (Byrnes *et al*., 2015a). Many businesses are also reluctant to grant external parties ongoing access to their systems (Byrnes *et al*., 2015a). However, external auditors may still leverage the benefits of continuous auditing by relying on the work of internal auditors to provide audit evidence (Teeter, 2014). External audit firms also benefit when they provide outsourced internal audit services (Byrnes *et al*., 2015a).

Even though there were no corresponding increases in the external audit environment, Byrnes *et al.*, (2015a) concluded that noteworthy gains were made by internal auditors in this field. However, industry studies, as discussed in paragraph 2.2, confirm that the efficient use of continuous auditing remains the biggest development area for internal audit functions.

3.5 Comparison of traditional and continuous auditing methodologies

Advances in accounting information systems, particularly ERP systems, have enabled real-time financial reporting (Chan & Vasarhelyi, 2011). Traditional audit methodologies have however not necessarily developed parallel to such real-time technology and economic environments (Chan & Vasarhelyi, 2011). Due to the manual nature of traditional audit procedures, such as the review of manual reconciliations, sampling and manual document verification, the frequency of audits is often limited to annual or bi-annual internal audit reviews. As a result, material errors may not be detected until the periodic (e.g. annual) internal audit is conducted (IIA, 2015). However, management and stakeholder reliance on real-time financial information is dependent on real-time assurance (Byrnes *et al*., 2015a). In the absence of real-time assurance, adverse management decisions could be made when using unaudited information. Therefore, the traditional audit process should be amended to support real-time assurance. Continuous auditing can be considered as a pro-active rather than a reactive audit methodology and is

therefore considered to be a successor of the traditional audit strategies (Chan & Vasarhelyi, 2011).

The most notable difference between traditional auditing and continuous auditing is the level of automation of audit procedures. Although data analyses may be utilised in traditional auditing, these analytical procedures are ad hoc in nature and not necessarily automated, as discussed in paragraph 3.6 (Chan & Vasarhelyi, 2011). Traditional and continuous auditing methodologies are compared in Table 3.1.

Table 3.1 Comparison of traditional and continuous auditing methodologies

	Traditional Auditing	Continuous Auditing
Frequency of testing and reporting	Periodic, e.g. annual	Real-time or frequent, e.g. weekly
Approach	Reactive	Pro-active
Procedure	Manual	Automated
Role of auditor	The majority of the audit work consists of time- and labour-intensive audit procedures	Consists of the investigation of exceptions and procedures requiring human judgement
Nature	Audit procedures mostly consist of analytical review procedures and substantive testing	Testing consists of continuous control monitoring and continuous data assurance
Timing	Controls testing and detailed testing occur separately	Controls monitoring and detailed testing occur simultaneously
Extent	Sampling is used extensively in testing transactions	Whole population is subject to testing
Resource	Manual execution of testing	Data modelling and analytics are used for monitoring and testing

(Source: Chan & Vasarhelyi, 2011)

3.6 The relationship between data analysis and continuous auditing

The effective use of data analysis is a precursor to implementing technology-enabled continuous auditing methodologies (IIA, 2011). Data analytics involves processes designed to obtain and evaluate data to extract and derive information for further use (IIA, 2011).

Data analysis as used by auditors refers to the process of identifying, gathering, validating, analysing and interpreting various forms of data (IIA, 2011). When data analysis is conducted, the overall objective and scope of an audit does not change. Data analysis is merely an alternative method to manual procedures which can be used to achieve the audit objectives (IIA, 2011). The results of data analytics may be used to identify areas of key risk, fraud, errors or misuse, improve business efficiencies, verify process effectiveness and influence business decisions (ISACA, 2011).

Technology-based audit tools which could be utilised for data analysis includes generalised audit software, spreadsheet software or scripts developed using audit-specific software, specialised audit utilities, commercially packaged solutions and custom-developed production systems (IIA, 2015). These audit tools form the foundation for continuous auditing. Technology-based audit tools are discussed further in paragraph 4.5.3.

Although data analysis is considered a precursor for continuous auditing (IIA, 2011), the implementation of data analysis technologies does not imply that continuous auditing is also implemented. Considering the activity-based maturity assessment discussed in paragraph 4.2.4, the initial phases of data analytics, namely ad hoc analytics, applied analytics and managed analytics, are not considered to be continuous auditing until a high degree of automation is achieved (KPMG, 2013), as explained below.

- **Ad hoc analytics** is the least mature level and is characterised by the basic use of analysis tools. Analytics are typically descriptive in nature and are limited to statistical analysis, classifications or summarisation of data. Ad hoc analytics are difficult to repeat in the absence of a standard methodology and documentation (IIA, 2011).
- The **applied analytics** level is characterised by integrating analytics into the audit processes (ISACA, 2011). Analytics are mainly used during audit fieldwork. It may also be used in the development of the audit plan, e.g. identifying financial statement trends (KPMG, 2013).
- The **managed analytics** level presents a controlled approach. Data, audit procedures and results are typically retained centrally, while standards for analytical procedure development are documented and analytical applications are executed against centralised data (IIA, 2011; ISACA, 2011).

- At the **automated analytics** level, protocols have been implemented for the automation of analytical procedures (IIA, 2011). Analytical procedures are considered repeatable at this level as the analytics logic is captured within program scripts (IIA, 2011; ISACA, 2011). Automated analytics is the first level of maturity that can be classified as **continuous auditing** (KPMG, 2013).

3.7 The elements of continuous auditing

Continuous auditing is broadly defined as the ongoing assessment of risks and controls which is achieved through automation, as discussed in paragraph 3.2 (Bumgarner & Vasarhelyi, 2015). Bumgarner and Vasarhelyi (2015) have however clarified this definition of continuous auditing by differentiating between four elements (refer to Figure 3.1). These elements are discussed in the remainder of this section.

Figure 3.1 The elements of continuous auditing

(Source: Bumgarner & Vasarhelyi, 2015)

3.7.1 Continuous data auditing

Internal auditors are faced with an expanding scope of activities while resources often remain limited (Soileau *et al*., 2015). This has contributed to the increased use of ad hoc transactional analytics as part of the traditional auditing methodology (Soileau *et al*., 2015). Examples of transactional analysis, which can be conducted continuously include:

- Extracting purchase transactions exceeding authorised limits;
- Summarising credit card transactions to identify excessive usage; and
- Comparison of account balances to the previous year (IIA, 2015).

The first implementations of continuous auditing were initially limited to the ongoing monitoring of transactions and exception-reporting mechanisms (Vasarhelyi & Halper,

1991). The initial concept of continuous auditing, i.e. transactional analysis and exception reporting, is now rather classified as **continuous data auditing** (Bumgarner & Vasarhelyi, 2015).

3.7.2 Continuous control monitoring

The initial scope of continuous (data) auditing was subsequently expanded to assurance on the adequacy of controls, in addition to only conducting transactional analysis (Alles *et al.*, 2006). Although similarly named, this element of continuous auditing should not be confused with the continuous monitoring activities of management (refer to paragraph 3.8).

Bumgarner and Vasarhelyi (2015) defined this element as **continuous control monitoring**. Alles *et al*. (2006) examined an audit approach which was developed in response to the Sarbanes-Oxley Act of 2002. Typical continuous control monitoring evaluates configurable controls against a baseline standard to identify any subsequent changes for further evaluation (IIA, 2015). Teeter (2014) extended this original work by examining a larger set of configurable controls of an ERP system. Configurable controls could include IT general controls, automated application controls, program changes and security parameters (IIA, 2015).

Examples of continuous control monitoring by internal audit functions include (IIA, 2015):

- Evaluating application configuration changes by comparing the current configuration setting to a baseline standard;
- Identifying program and parameter changes for further evaluation;
- Scanning operating systems for patch levels; and
- Analysing incident and error management systems for risk indicators.

An audit planning framework to implement this element of continuous auditing, together with practical continuous audit (control monitoring) procedures for database management systems, is the focus of this study.

3.7.3 Continuous risk monitoring

Vasarhelyi *et al*. (2010) suggested the addition of **continuous risk monitoring** to the continuous auditing schema. Internal audit functions judgementally select risks for

monitoring against key risk indicators to detect significant changes in risk. These monitoring activities should be automated, similar to the other elements of continuous auditing. Any increases or changes in the risk indicators are considered for inclusion in the audit plan, or alternatively, communicated to management (IIA, 2015). For example, an increase in IT security incidents could be a leading indicator of a system compromise (Byrnes, Brennan, Vasarhelyi & Moon, 2015c).

3.7.4 Continuous compliance monitoring

In response to the increase in legal and regulatory compliance requirements of the modern business world, Bumgarner and Vasarhelyi (2015) propose that **continuous compliance monitoring** be added as the fourth element of continuous auditing. Continuous data, controls and risk monitoring are complementary to continuous compliance monitoring and may have shared design, analytical and technology components (Bumgarner & Vasarhelyi, 2015).

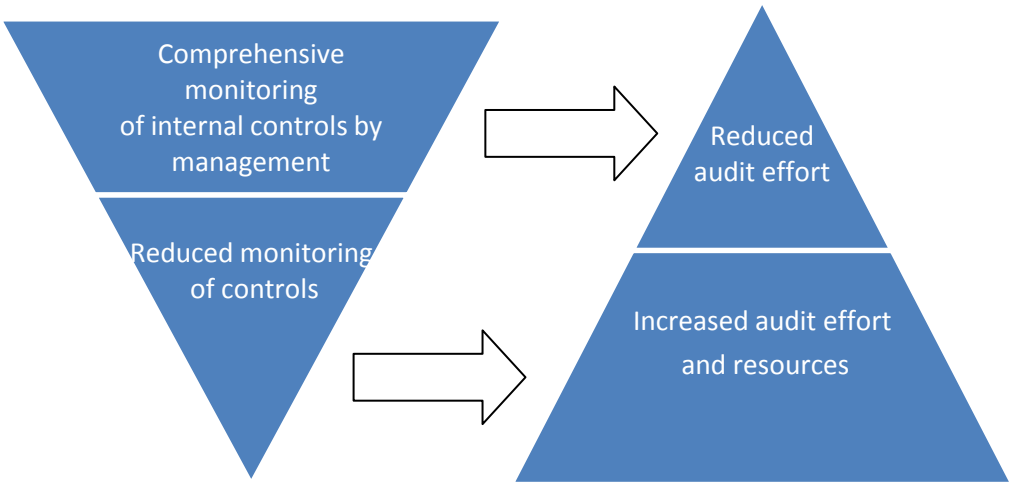
3.8 Continuous monitoring

Continuous monitoring is a process performed by management to monitor on an ongoing basis whether internal controls are operating effectively (IIA, 2015). Many of the techniques employed by management to monitor controls continuously are similar to continuous auditing techniques used by internal auditors (IIA, 2015).

Continuous monitoring allows an organisation to observe one or many processes, systems or types of data. Similar to executive information systems, continuous monitoring systems are designed to generate summary information such as daily sales volumes and billing. Other examples are the monitoring of accounts payable and cash disbursement activities, including identifying duplicate transactions by comparing reference numbers, account numbers and amounts (ISACA Standards Board, 2002).

There is an inverse relationship between continuous auditing and continuous monitoring performed by management, as depicted in Figure 3.2. Internal auditors should adjust the extent of continuous auditing work based on the adequacy of management's continuous monitoring processes. Should the continuous monitoring process be inadequate, auditing efforts should increase accordingly (IIA, 2015).

Figure 3.2 The relationship between continuous auditing and continuous monitoring



(Source: IIA, 2015)

Since continuous monitoring procedures performed by management are often similar to those continuous auditing procedures performed by internal auditors, internal auditors should ensure that they do not retain ownership for continuous monitoring activities as this could be presumed to impair the independence of the auditor (IIA, 2015).

Auditing standards (e.g. IIA Practice Advisory 2320-4, ISACA Standard 1002-3) state that the monitoring of processes, systems and data forms part of management's responsibility to implement and maintain an effective control environment (ISACA, 2014; IIA, 2013a). Therefore, internal audit functions should refrain from assuming a monitoring role under the auspices of continuous auditing (ISACA Standards Board, 2002).

Information provided by a continuous monitoring system can provide internal auditors with information about a process, system or data (ISACA Standards Board, 2002). The internal auditor's objective is to accumulate independent audit evidence to reduce the audit risk to an appropriate level (ISACA Standards Board, 2002). Due to the indirect nature of information provided by a continuous monitoring system, this information cannot be utilised as audit evidence without corroborating the information with directly obtained evidence (ISACA Standards Board, 2002). Additional independent procedures are therefore required to corroborate continuous monitoring activities (ISACA Standards Board, 2002).

3.9 Continuous assurance

Continuous assurance is a combination of the internal auditor's continuous auditing processes and audit testing of continuous monitoring activities performed by financial, operational and IT management, as depicted in Figure 3.3 (Bumgarner & Vasarhelyi, 2015). The auditor should examine the adequacy of management's continuous monitoring activities to determine whether the auditor can reduce the detailed testing of controls (IIA, 2013a; KPMG, 2013).

Figure 3.3 Continuous assurance

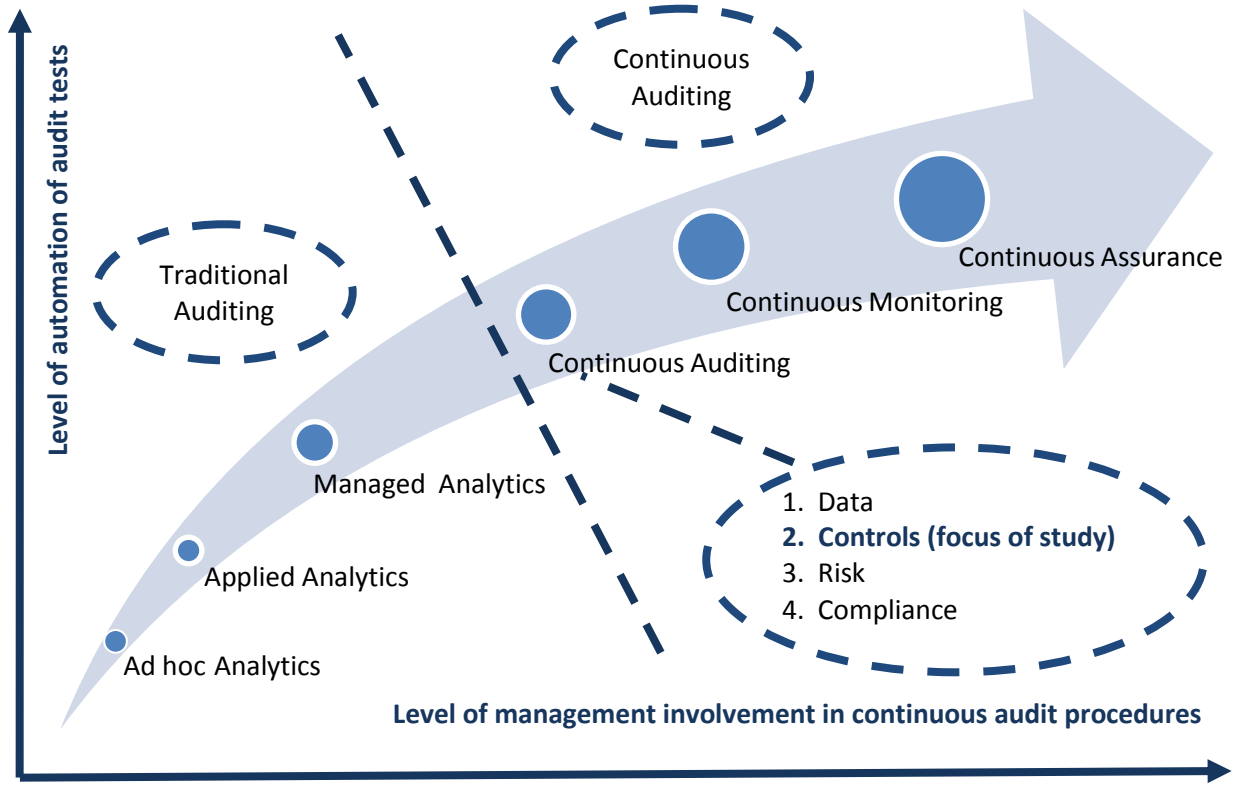
(Source: Bumgarner & Vasarhelyi, 2015)

As continuous auditing aims to establish whether policies and controls are operating effectively, audit procedures are also extended to the continuous monitoring processes implemented by management, resulting in continuous assurance (IIA, 2015; Roth, 2012).

3.10 Conclusion

The term continuous auditing is often used interchangeably with related concepts such as data analytics, continuous monitoring and continuous assurance. As a result, academics and auditing standards setters continue to refine and re-define the concept, definition and elements of continuous auditing. For the purposes of this study, the definition of continuous auditing is consistent to that of the IIA (2015): *The combination of technology-enabled ongoing risk and control assessments.* The evolving nature of the continuous auditing process and the related topics, data analytics and continuous monitoring is depicted in Figure 3.4.

Figure 3.4 The evolution from traditional auditing to continuous auditing



(Sources: IIA, 2015; KPMG, 2013; IIA, 2011; ISACA, 2011)

Implementing a continuous auditing process is typically preceded by the inclusion of ad hoc data analytics during the execution of audit fieldwork (KPMG, 2013). Although applied and managed analytics have a higher degree of automation, these precursors of continuous auditing are still classified as traditional auditing (KPMG, 2013). These have the potential to evolve to continuous auditing, by implementing repeatable and managed analytical processes (KPMG, 2013). As the levels of automation and management involvement increases, the continuous auditing initiatives may mature to reach the ultimate level of maturity, namely continuous assurance (Bumgarner & Vasarhelyi, 2015).

The modern definition of continuous auditing consists of four elements, namely continuous **data auditing, control monitoring, risk monitoring** and **compliance monitoring** (Bumgarner & Vasarhelyi, 2015). Transactional data analysis such as isolating outlier transactions and measuring changes in internal indicators (e.g. number of high value transactions) and external indicators (e.g. macro-economic factors) over time is used to provide assurance using continuous data auditing, risk monitoring and compliance monitoring (IIA, 2015). These elements of continuous auditing are excluded from the

scope of this study, which focuses on the **continuous control monitoring** element only. The continuous control motoring element provides continuous assurance over the adequacy and effectiveness of automated IT controls (IIA, 2015).

Considering that the implementation of continuous auditing techniques and its precursor, data analytics, is still immature in practice (Gonzalez *et al*., 2012), this study will focus on practical guidance in order to advance the implementation of continuous auditing techniques, as discussed in Chapter 2. While continuous assurance may be the ultimate objective for internal audit practitioners, this level of maturity cannot be achieved if the foundational levels are not in place (IIA, 2015). Therefore, this study focuses on one such foundational element, namely continuous controls monitoring of automated IT controls.

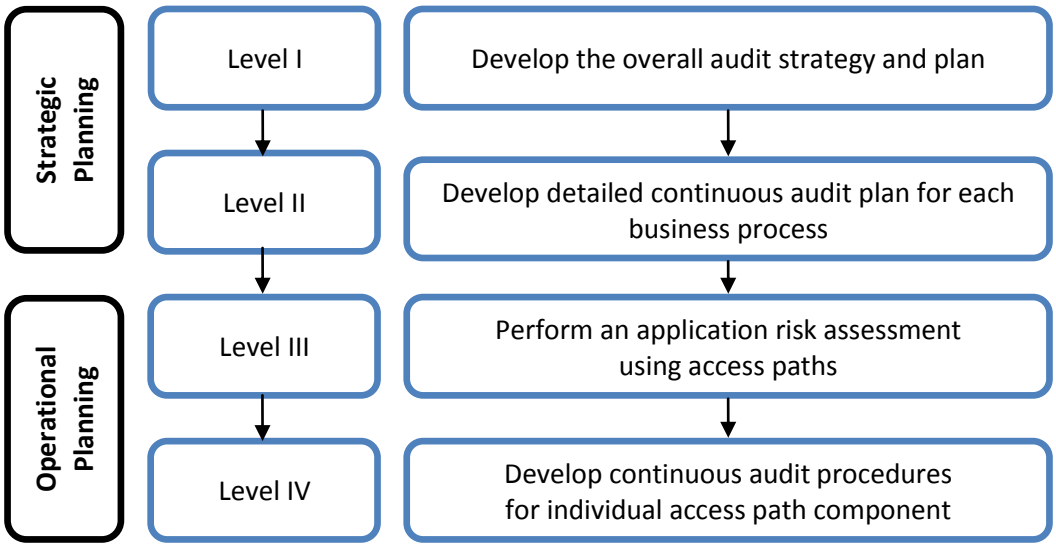
When planning for the implementation of continuous auditing procedures, the four elements of continuous auditing should be considered. While implementation guidance relating to **continuous data, risk** and **compliance monitoring** is generally available, limited literature is available relating to the **continuous control monitoring** element of continuous auditing. As such, an audit planning framework for implementing continuous auditing techniques, which focuses on the continuous control monitoring for automated IT controls, is discussed in Chapter 4.

CHAPTER 4. FINDINGS: AUDIT PLANNING FRAMEWORK AT A STRATEGIC AND OPERATIONAL LEVEL FOR IMPLEMENTING CONTINUOUS AUDITING

4.1 Introduction

Planning for continuous auditing is not conducted in isolation from planning traditional audit procedures, but is rather embedded in the annual audit planning process (IIA, 2015). In this chapter, an audit planning framework for implementing continuous auditing is developed. This framework consists of planning at a strategic and operational level, as shown in Figure 4.1.

Figure 4.1 Continuous auditing planning levels



(Source: Author's own construct)

The implementation of continuous auditing is planned firstly at a strategic level when developing the overall continuous auditing strategy and the audit plan. Once the strategy is determined, planning is conducted at an operational level for each business process identified for continuous auditing. In Chapter 5, this framework is applied practically by developing continuous auditing procedures for database management systems.

Strategic audit planning typically commences with developing an audit universe and performing a business risk assessment at macro-level (Level I – refer to paragraph 4.2). It

is recommended that a maturity assessment relating to continuous auditing initiatives is performed as part of strategic planning. Annual planning is followed by a more detailed business process risk assessment which would include the development of a continuous auditing implementation plan (Level II – refer to paragraph 4.3).

At an operational level, the framework is intended to support the planning of detailed procedures for the controls monitoring component of continuous auditing. In order to identify automated IT controls for continuous auditing purposes, the business process risk assessment is further broadened by performing an application risk assessment and analysing IT access paths (Level III – refer to paragraph 4.4). Continuous auditing procedures are finally developed for each access path component, depending on the lifecycle phase of the product's development (Level IV - refer to paragraph 4.5).

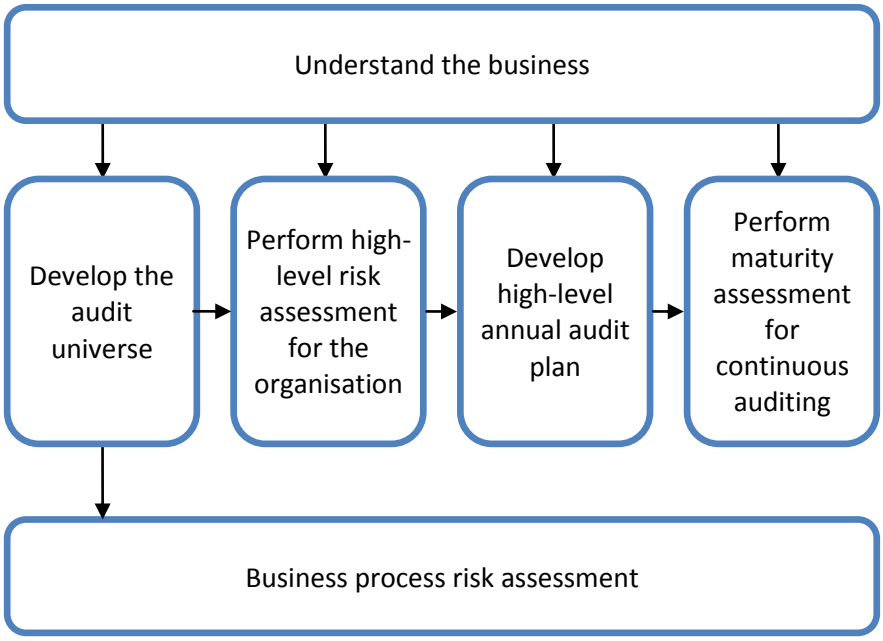
4.2 Level I: Develop the overall audit strategy and plan

The implementation of continuous auditing should commence with developing a strategy which considers the current and desired states of maturity for continuous auditing initiatives of the particular organisation, as shown in Figure 4.2 (IIA, 2015). Annual audit planning should commence with developing a register of business processes in combination with a risk assessment, to identify the potential areas in which to possibly perform such procedures (IIA, 2013a).

ISACA and IIA auditing standards require internal audit functions to define an annual internal audit plan by following a systematic process, which ensures that all fundamental business processes and IT services are included (IIA, 2013a; ISACA, 2014). This process includes liaising with operational management as well as the compliance and risk management functions to encourage support of the continuous auditing strategy (IIA, 2015). As part of the strategic planning process, the overall audit plan should be adjusted to reflect the changes introduced by continuous auditing (IIA, 2015).

However, neither of the IIA (2013a) or the ISACA (2014) auditing standards prescribes a specific annual audit planning process to be followed. Therefore, for the purposes of this study, the overall IT audit planning methodology described by the IIA (2008) and the continuous audit planning methodology proposed by Corderre (2010) were used in this research and are combined below (shown in Figure 4.2).

Figure 4.2 Level I: Develop the overall audit strategy and plan



(Sources: IIA, 2008; Corderre, 2010)

4.2.1 Develop the audit universe

The audit plan should be based on the internal auditor's understanding of the business, including the objectives, strategies and business model that will enable the understanding of the organisation's unique business risks (Corderre, 2010).

Once sufficient knowledge of the business is gathered, the audit universe should be documented (IIA, 2008). The audit universe is a register of audit areas which serves as the source from which the annual audit plan is prepared (ISACA, 2016). The audit universe is developed by identifying key business objectives and processes, the software applications which support the key processes, the IT infrastructure required by the business applications, and the organisation's IT service support model (IIA, 2008).

The IT audit universe should be embedded in the overall audit universe due to the strong interdependencies of IT and the business processes that it supports (IIA, 2013b; Corderre, 2010).

4.2.2 Perform high-level risk assessment

After the audit universe is documented, a risk assessment should be performed for each identified component of the audit universe (IIA, 2008). This entails determining the impact and likelihood of each event which could hinder the organisation from attaining its business objectives in an effective, efficient and controlled manner (IIA, 2008).

4.2.3 Develop high-level annual audit plan

The next step is to formulate the audit plan, focusing the auditor's assurance activities on those areas which could provide management with objective information to manage the organisation's business risks (IIA, 2008). Based on the risk assessment and the level of automation of the internal controls, this process will include identifying areas suitable for continuous auditing (IIA, 2015).

Planning for continuous auditing activities is not conducted in isolation from the annual audit planning process (IIA, 2015). When developing the audit plan, continuous auditing could assist the internal auditor in compiling an audit plan which is responsive to changes in business risks (IIA, 2015). In particular, instead of scheduling audits according to a rotational cycle of coverage (e.g. six-monthly, annually, every second year), the frequency of audits should rather be determined based on risk, complexity, pervasiveness and velocity of change (IIA, 2015). Continuous data analytics should include leading indicators to indicate specific processes or systems for traditional or continuous auditing (IIA, 2015).

The potential areas where continuous auditing is to be implemented are therefore identified as part of the overall audit plan development. However, prior to implementing or enhancing continuous auditing, the adopting organisation should perform a maturity assessment of the existing data analytics and continuous auditing efforts to assist in planning the implementation process (KPMG, 2013).

4.2.4 Perform maturity assessment for continuous auditing activities

The level of maturity of existing data analytics and continuous auditing initiatives should be assessed to aid the implementation steps in reaching the desired state of maturity (KPMG, 2013). KPMG (2013) recommends that audit functions should use an audit methodology-based maturity assessment model, as depicted in Table 4.1.

The maturity assessment model assesses the capabilities of the data analytics and continuous auditing initiatives deployed by the internal audit function (IIA, 2015). The optimisation of such initiatives is measured for five aspects of the audit process, namely strategic analysis, enterprise risk management, audit plan development, fieldwork and reporting, as well as continuous improvement (KPMG,2013).

It is recommended that the maturity assessment be performed as part of strategic and annual audit planning, prior to developing a detailed implementation plan for continuous auditing (KPMG, 2013). Continuous auditing initiatives should also be evaluated periodically to refresh the overall strategy and to identify additional controls and parameters to be tested (IIA, 2015).

Table 4.1 Level I: Audit methodology-based maturity assessment model

		Level of maturity of data analytics and continuous audit initiatives					
		Ad hoc Analytics	Applied Analytics	Managed Analytics	Continuous Auditing	Continuous Monitoring	Continuous Assurance
Audit process	Strategic analysis	○	○	○	◎	◎	●
	Enterprise risk management	○	○	○	◎	◎	●
	Audit plan development	○	○	◎	◎	●	●
	Fieldwork and reporting	◎	◎	◎	●	●	●
	Continuous improvement	○	○	○	○	◎	●
	Type of data analytics	❶	❶	❷	❸	❹	❹
		Traditional Auditing			Continuous Auditing		

○	Data analytics are generally not used	◎	Data analytics are partially used/sub-optimised	●	Data analytics are optimised (effective and consistent)		
❶	Descriptive	❷	Descriptive Diagnostic	❸	Descriptive Diagnostic Predictive	❹	Descriptive Diagnostic Predictive Prescriptive

(Sources: IIA, 2015; KPMG, 2013; IIA, 2011; ISACA, 2011)

When conducting the maturity assessment depicted in Table 4.1, the maturity level each of the five aspects of the audit process should be assessed, by considering the extent and level of integration of analytical capabilities for the specific process (KPMG, 2013).

Similarly, data analytics initiatives can be categorised as descriptive, diagnostic, predictive or prescriptive (KPMG, 2013). Analytical capabilities typically progresses from being descriptive only, on the least mature end of the scale, to the most mature state that includes all four types of analytics (KPMG, 2013).

At the least mature level, data analytics initiatives are mainly descriptive in nature and only provide information about the trends, patterns and relationships in existing data (KPMG, 2013). As maturity increases, data analytics include diagnostic procedures to understand the underlying cause of a particular result (KPMG, 2013). Analytics may also be predictive in nature by using historical data to forecast a predicted outcome for new data sets (KPMG, 2013). At the most mature level, analytics also includes prescriptive procedures to articulate the ideal process to follow in response to an event (KPMG, 2013).

For example, audit methodologies which deploy data analytics only during the fieldwork and reporting phases of an audit, are classified at the lowest level of maturity, namely ad hoc analytics, which is still considered to be traditional auditing (KPMG, 2013). The data analytic capabilities are mainly descriptive in nature at this level (KPMG, 2013). Only once data analytics are optimised across the entire audit process, the continuous assurance level have been reached (KPMG, 2013). At the ultimate level of maturity, analytics are descriptive, diagnostic, predictive and prescriptive in nature (KPMG, 2013).

4.3 Level II: Develop a continuous audit implementation plan for selected business processes

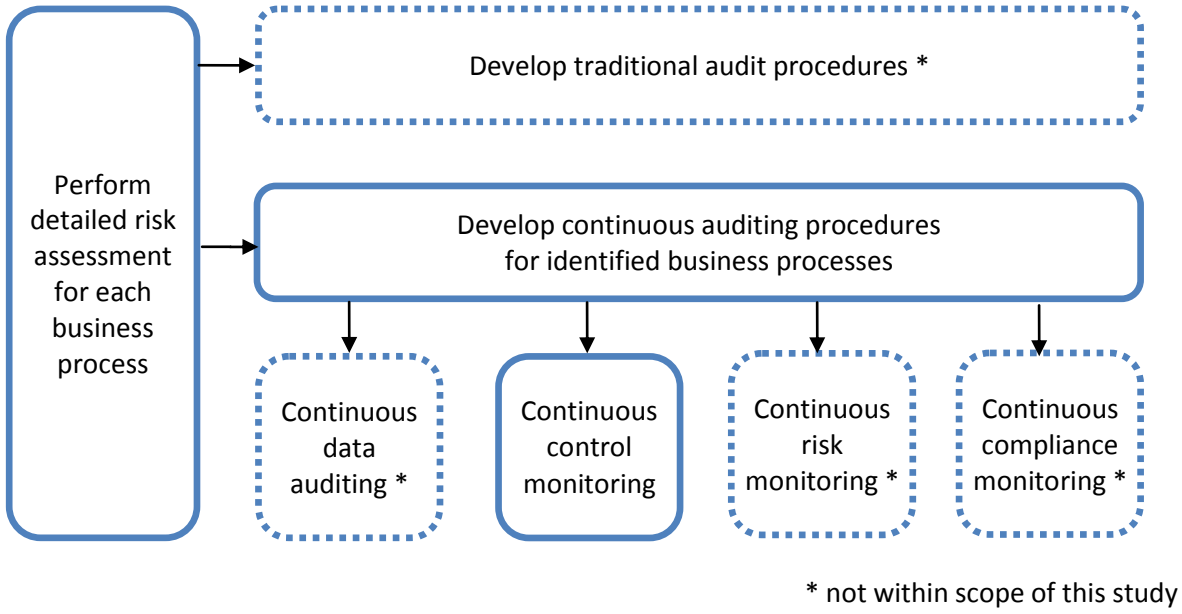
Based on the overall risk assessment, audit plan and maturity assessment (Level I), the internal audit function should develop a road map for continuous auditing which is integrated with the annual audit plan (Schultz, 2014). This commences by conducting a risk assessment and identifying key business objectives for each business process. The IIA (2015) suggests that the audit procedures are adapted to specify ongoing risk and control indicators. For the processes where continuous auditing techniques are not considered feasible, the traditional audit methodologies will apply (IIA, 2015).

To determine whether continuous auditing methodologies could be implemented, the auditor should design specifications for ongoing risk indicators and control measurements (IIA, 2015). The ongoing risk indicators and control measurements should consider the

four elements of continuous auditing described in paragraph 3.7, namely continuous data auditing, control monitoring, risk monitoring and compliance monitoring (Bumgarner & Vasarhelyi, 2015).

Internal auditors and business management should collaborate to determine the leading and lagging indicators that measure the risks and controls related to the particular business objectives (IIA, 2015). Schultz (2014) recommends an integrated approach, which involves the operational, financial and IT audit teams, as well as management, to ensure that all risks and controls are considered. Continuous data auditing, risk monitoring and compliance monitoring are mainly conducted through the implementation of transactional data analysis such as isolating outlier transactions and measuring changes in key indicators over time (IIA, 2015). These elements of continuous auditing are excluded from the scope of this study, which focuses on the **continuous control monitoring** element only. Also refer to Figure 4.3.

Figure 4.3 Level II: Evaluate business processes for continuous auditing



(Sources: IIA, 2015; Bumgarner & Vasarhelyi, 2015)

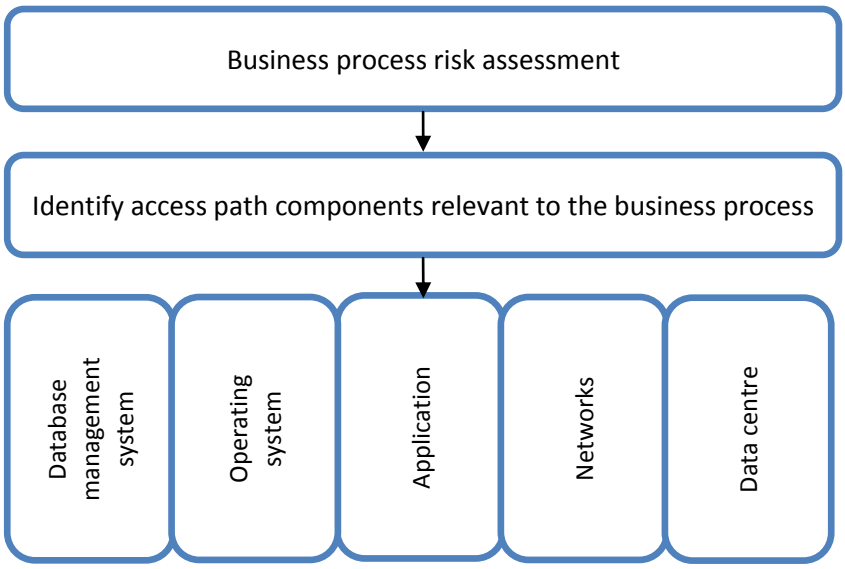
The continuous control monitoring element of continuous auditing provides an independent review of the configured application controls and IT general controls by evaluating whether controls are adequate and effective, followed by the ongoing identification of changes to those controls (IIA, 2015). To identify the configuration controls relevant to the particular business process under review, the IT architecture components underlying that business

process should be identified through IT access path analysis to aid the risk assessment (Boshoff, 2014).

4.4 Level III: Perform an application risk assessment using access paths

The documentation of an IT audit universe entails, among other things, compiling a comprehensive inventory of the organisation's IT architecture components (IIA, 2008). This is done by first identifying those business processes supporting strategic objectives and then identifying the underlying technical infrastructure, including the application's program code, database, operating system and network infrastructure (Cascarino, 2012). This is referred to as the multi-tier model (Gibbs, Jain, Joshi, Muddamsetti & Singh, 2010), as shown in Figure 4.4.

Figure 4.4 Level III: Multi-tier model to identify IT architectural components



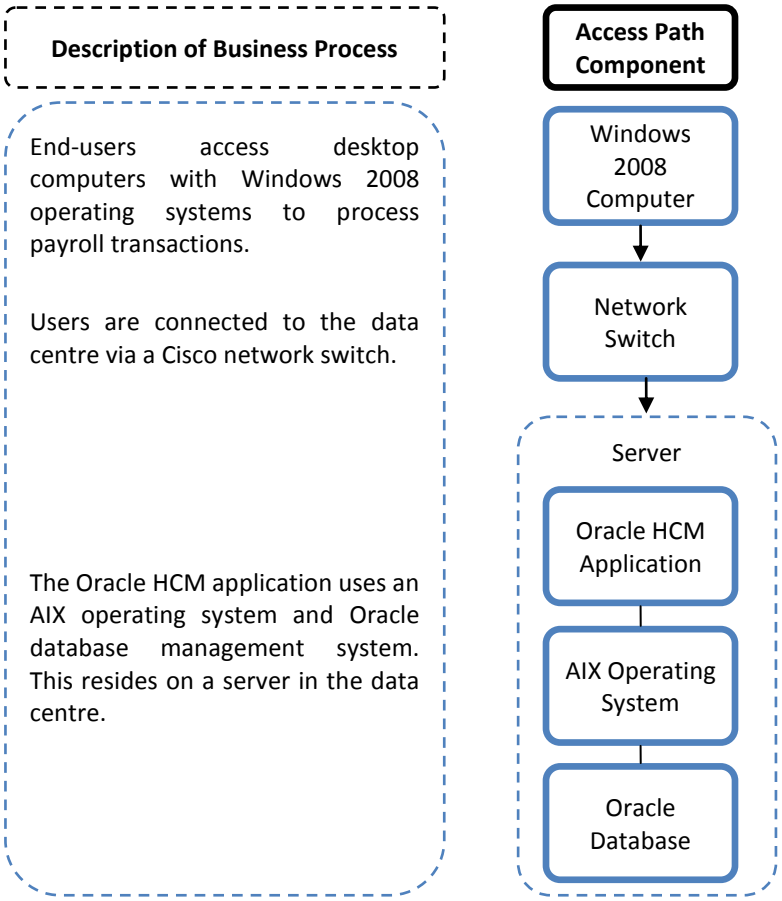
(Sources: Cascarino, 2012; Davis *et al*., 2011; Gibbs *et al*., 2010; IIA, 2008)

Similar to the multi-tier model, the "access path" model was developed by Boshoff (1990). This model ensures that all possible access points, relating to each of these IT architecture components, are identified. An access path is the logical route which is activated when an end-user accesses computer-controlled resources (Boshoff, 1990). An end-user may pass through one or multiple IT architectural components before obtaining access to the data resources (Goosen, 2012). Access paths typically include the personal computers, networking equipment (routers and switches) and application packages, operating systems and databases (Boshoff, 2014), as described in Appendix 1.

Since there may be multiple access paths for the same activity, the auditor should identify all possible access paths to the data resources to assess the risks and configurable controls associated with each IT component which could be activated during this process (Cascarino, 2012). This includes direct or "backdoor" methods of accessing data by operators and programmers (Cascarino, 2012).

A simplified example of an access path schematic is depicted in Figure 4.5 for an end-user who processes payroll transactions on an Oracle Human Capital Management (HCM) application.

Figure 4.5 Level III: Simplified example of an IT access path



(Source: Boshoff, 2014)

All the components of an IT access path are essential to enable automated business functionality and such components pose risks relating to the availability, integrity and confidentiality of data (IIA, 2008). The degree of risk is influenced by the criticality of the business activity supported by the technology, and on the technology's configuration settings (IIA, 2008). Therefore, the larger the variety of access paths for each of these

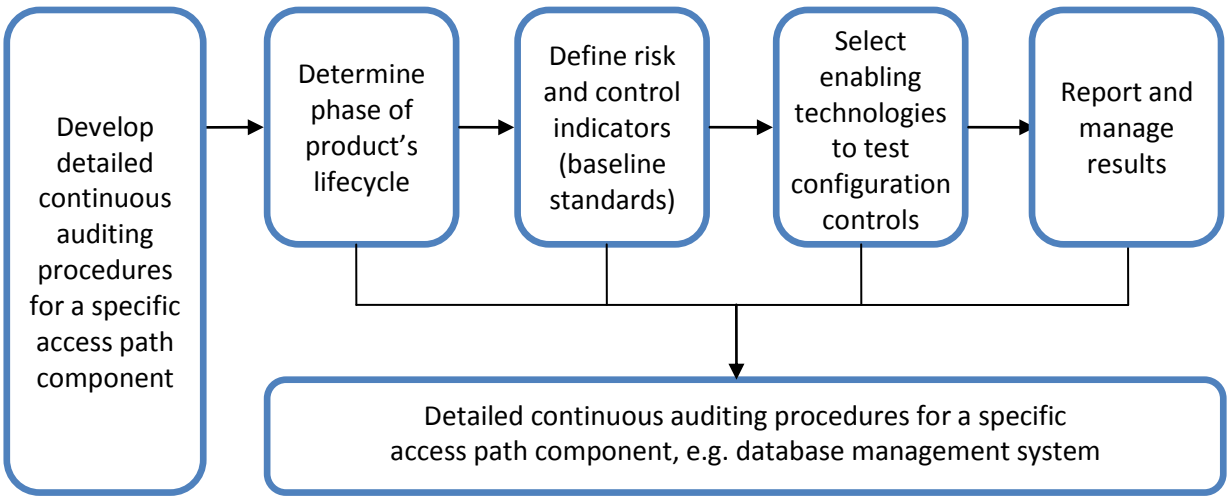
components, the higher the organisation's risk profile relating to unauthorised access to data (IIA, 2008).

4.5 Level IV: Develop continuous audit procedures for individual access path components

Once the IT access path components relevant to the business process under review have been identified, detailed audit procedures should be developed, depending on the particular lifecycle phase for each access path component. The product lifecycle consists of a series of stages, such as either being **build, set-up, configured, maintained** or **operated** (Boshoff, 2014).

The lifecycle phase influences the selection of configuration controls (baseline risk and control indicators) to test continuously (Boshoff, 2014; Cooke, 2014). Considering the particular access path components and the baseline control standard, the auditor will then select the appropriate software tools to be used for continuous auditing purposes (IIA, 2015), as shown in Figure 4.6.

Figure 4.6 Level IV: Develop continuous auditing procedures for selected access path component



(Source: Author's own construct)

4.5.1 Determine the product's lifecycle phase

Goosen (2012) developed an integrated IT governance framework which provides guidance for implementing configuration controls at an operational level for each access path component considering the specific lifecycle phase. Once all access path components for the particular business process have been identified, the identified components should be examined by the auditor to ensure that it is correctly built, set up, configured, maintained and/or operated through-out the product's lifecycle in such a manner to mitigate the associated risks (Goosen & Rudman, 2014; Boshoff, 2014). Each lifecycle phase is described in Table 4.2.

Table 4.2 Description of product lifecycle phases

Lifecycle Phase	Description
Build	The build phase is the process of assembling computer hardware components to accept an operating system. Computer software could also be built by creating and converting source code files into stand-alone software artefacts, which can be executed on a computer. This includes the conversion of source code files to executable code.
Set-up	Set-up refers to the installation of a software program on a computer system.
Configure	The initial settings of computer programs are configured according to the particular business requirements. Configurable items include applications, server processes and operating system settings.
Operate	A computer or system is operated by overseeing the running of the computer. Computer operations include the stopping and restarting of selected services or the whole computer or system.
Maintain	The maintain phase includes software upgrades and computer/hardware repairs to ensure the optimum performance and reliability of the device.

(Source: Boshoff, 2014).

Boshoff (2014) proposed a framework which links the identified IT access path components with the applicable product lifecycle phase, as referred to in Figure 4.7. The auditor firstly determines which lifecycle phase is applicable to the particular access path component under review. Thereafter, audit procedures are developed to address the risks particular to that access path component, considering the business risks and lifecycle phase. This framework will be used to develop continuous auditing procedures for database management systems, as discussed in Chapter 5.

Figure 4.7 Level IV: Identify the lifecycle phase for access path components

		Determine applicable lifecycle phases				
Access path component	**Example**	**Build**	**Set up**	**Configure**	**Operate**	**Maintain**
Personal computer	Windows 2008 Computer	Maybe	Maybe	Yes	Yes	Yes
Network Switch	Cisco Network Switch	No	No	Maybe	No	Yes
Server	**Server**	Maybe	Maybe	Yes	Yes	Yes
Application Software	Oracle HCM Application	No	Yes	Yes	Yes	Yes
Operating system	AIX Operating System	No	Yes	Yes	Yes	Yes
Database	Oracle Database	No	Yes	Yes	Yes	Yes

(Source: Boshoff, 2014)

4.5.2 Define risk and control indicators (baseline standards)

Continuous audit risk and control indicators should be developed, consistent with the IIA standard 2120, to enable ongoing assessments in two dimensions, namely risk and control assessments. This should be done in collaboration with business management and IT professionals (IIA, 2015).

The **continuous risk assessment** dimension should identify increased levels of risk at a macro-analytics or strategic level, considering key metrics trends and patterns using transactional analysis (KPMG, 2015). This dimension is excluded from the scope of this study, which focuses on continuous control monitoring only.

The **continuous control assessment** dimension of continuous auditing entails identifying metrics for each business process, considering the underlying IT operations (general controls) and configured controls (IIA, 2015). This is consistent with the continuous control monitoring component of continuous auditing (Bumgarner & Vasarhelyi, 2015). For the remainder of this study, only this dimension of continuous auditing is discussed.

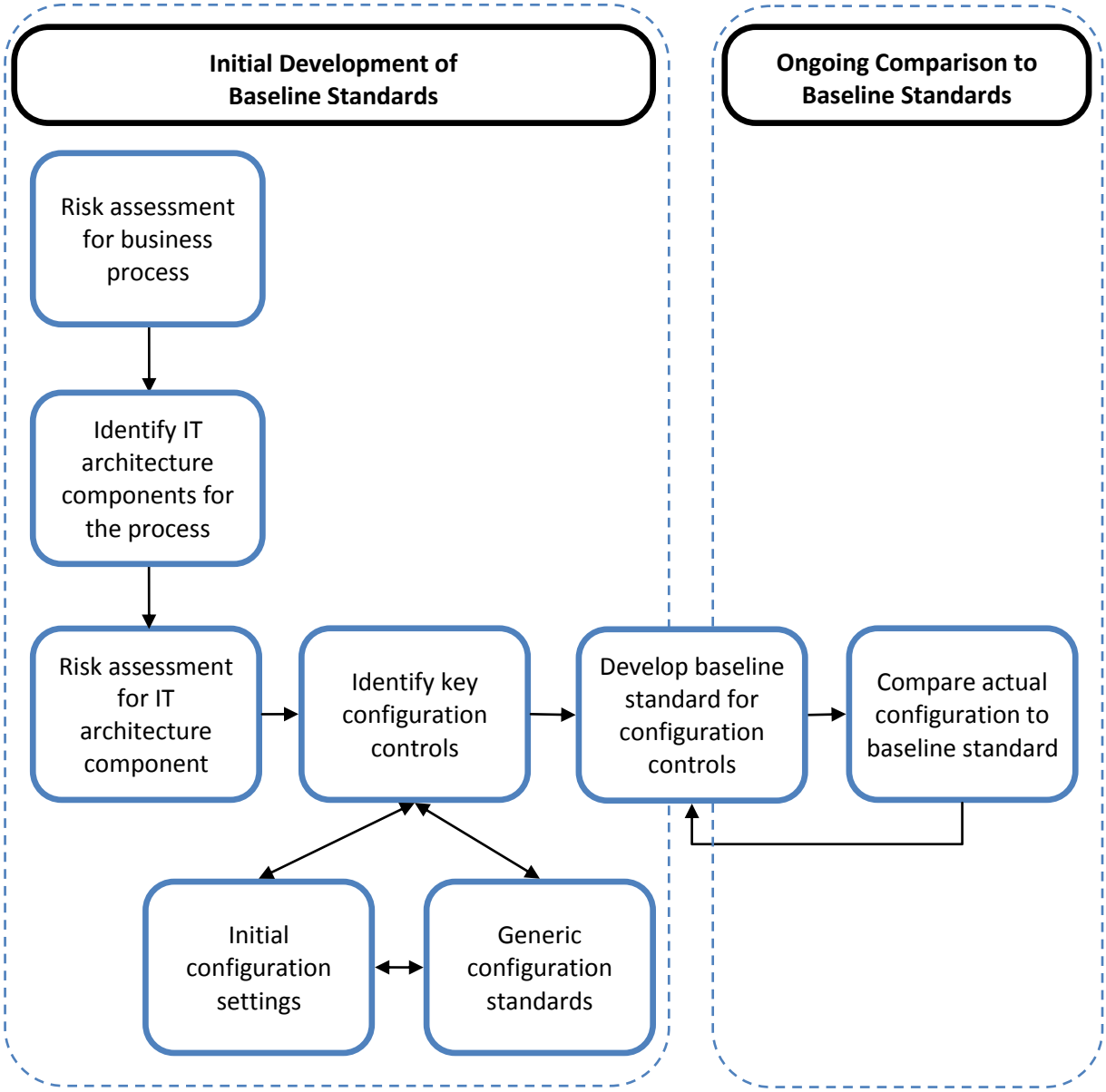
The IIA (2015) proposes a continuous control monitoring methodology for IT operations and applications (systems) that entails continuously assessing configured controls against a baseline condition to identify changes to configured controls, as shown in Figure 4.8. This concept is similar to the idea of Davis *et al*. (2011) which entails an initial review of the configuration settings of IT systems against the organisation's development standards. Future (i.e. continuous) reviews will entail identifying and reviewing any deviations from the baseline standard for such systems (Davis *et al*., 2011). An example of a baseline standard comparison is depicted in Table 4.3.

Table 4.3 Level IV: Example of a continuous baseline standard comparison

Configuration Setting/ Automated Control Examples	Baseline standard	Actual result	Audit status	Identified for audit?
Number of failed login attempts permitted	3 attempts	99 attempts	Changed	Yes
Password complexity verification is enabled	Enabled	Enabled	Unchanged	No
Default passwords are noted	None	Yes	Failed	Yes
Logging is enabled	True	True	Unchanged	No
Limited users with privileged (administrator) access	User1 User2	User1 User3	New/deleted entries	Yes

(Sources: IIA, 2015; Teeter, 2014; Cooke, 2014)

Figure 4.8 Level IV: Continuous auditing using a baseline standard



(Sources: IIA, 2015; Tysiac, 2015; Teeter, 2014; Cooke, 2014)

The control assessment process commences with identifying the control objectives, such as validity, confidentially, integrity and availability, which is associated with the process and IT system under review (IIA, 2015). Key configurable controls are then identified through scenario analysis and system walk-through descriptions (IIA, 2015; Gibbs *et al*., 2010). Only key controls which address high-risk areas are tested to evaluate whether they are adequate in design and operating effectively. Once evaluated, these controls may then serve as the baseline standard of the control without detailed re-evaluation in

future (Tysiac, 2015). If the organisation has documented IT standards, these could also be used as the baseline standard (Cooke, 2014).

Once the baseline standard is determined, appropriate analytical procedures should be developed to identify any subsequent changes or deviations from the standard (IIA, 2015). In particular, the organisation's data analytics software is configured with this baseline standard (Cooke, 2014). Extracts of system configuration settings are then compared to the baseline standard using the data analytics software (Cooke, 2014). This comparison is repeated periodically to highlight changes to the confirmation settings (IIA, 2015). In this manner, non-compliant and changed configuration settings can be identified for review by the auditor and possible mitigation by management (IIA, 2015). Both general analytical software and commercial vulnerability assessment tools can be used to achieve this objective (Teeter, 2014).

A baseline control standard should be developed for each IT access path component. This baseline standard should ideally be determined when the system is implemented (i.e. the system is built, set-up and configured), whereafter any subsequent configuration changes to such IT components are identified for auditing during the production phase of the lifecycle (i.e. when the system is operated and maintained) (IIA, 2015). The data extraction and comparison process should be automated by using inter alia generalised audit software (IIA, 2015).

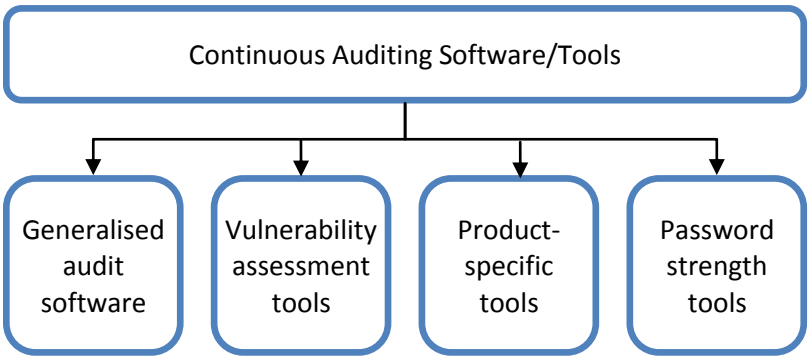
Once the relevant lifecycle phase of the selected access path component and the baseline standards for configuration controls have been identified, the auditor can commence with selecting the appropriate technology solution to automate the testing of these controls.

4.5.3 Audit software/tool selection

Audit software or tools to be used for specific continuous auditing procedures vary depending on the specific access path component under review as well as the particular lifecycle phase (IIA, 2011). Generalised audit software could be used to continuously test the configuration controls of the identified access path components at each applicable lifecycle phase (Cooke, 2014). Audit-specific analytical software is flexible and can read diverse data types, including mainframe legacy systems, client/server and Internet-enabled systems, or enterprise resource applications such as SAP and Oracle (IIA, 2015). Considering the emphasis on automation and technology in the definition of continuous

auditing (refer to paragraph 3.2), the selection of appropriate auditing tools is imperative to the continuous auditing methodology (Cangemi, 2015). Although this research study was aimed at developing a generic continuous auditing framework using generalised auditing software, four types of software will be discussed, as shown in Figure 4.9.

Figure 4.9 Level IV: Selecting audit software/tools



(Source: Author's own construct)

4.5.3.1 Generalised audit software

CAATTs are required to process the large volumes of data of a complex business environment (Cascarino, 2012). Generalised audit software is a sub-set of CAATTs which is specifically intended for data retrieval and analysis purposes by auditors (De Kroon & Karp, 2013). The software typically has features to organise, combine, extract, reformat and analyse data across multiple data sources and systems (Cascarino, 2012). It also has an audit trail function to log procedures performed by the auditor and has the ability to re-execute analysis with minor changes (De Kroon & Karp, 2013).

In a survey of data analytics software, the most prevalent generalised audit software used by auditors are ACL (Audit Command Language) (39% of respondents) and CaseWare Analytics/IDEA (16% of respondents) (AuditNet, 2015). Other software used by auditors are Microsoft Excel (70% of respondents) and Microsoft Access (25% of respondents) (AuditNet, 2015). Conventional programming languages such as SQL may also be an option should the auditor have the necessary skills at their disposal (Cascarino, 2012).

Since there are a variety of tools available, the most appropriate technology should be selected for the particular organisation's audit tasks, objectives and IT environment. This

should be done taking into consideration the overall risk assessment, among other things (IIA, 2011).

4.5.3.2 Generic vulnerability assessment tools

Vulnerability assessment tools are used by IT management to identify, categorise and manage security vulnerabilities such as unsecure system configurations or missing security updates of IT architecture components (Rochford & Akshay, 2015). This type of tool typically includes configuration auditing, target profiling, penetration testing and detailed vulnerability analysis for widely-used systems (Lindros & Tittel, 2014). According to Rochford and Akshay (2015), these tools typically have the following functionalities:

- Discover and identify network IT assets;
- Report the security configuration settings of IT assets and any changes thereto;
- Establish a baseline of vulnerability conditions for devices, applications and databases;
- Produce customised reports for specific compliance regimes, control frameworks and audiences.

This market is dominated by five vendors namely BeyondTrust, Rapid7, Tenable Network Security (Nessus), Tripwire and Qualys (Rochford & Akshay, 2015). Negligible differences have been noted between solutions, while purchase decisions are mostly based on cost (Rochford & Akshay, 2015).

Although there are a number of generic vulnerability assessment tools available, it is not always feasible to implement these tools. In particular, Cooke (2014) and Teeter (2014) point out the following:

- Commercial tools may be too costly for smaller organisations;
- Geographically dispersed organisations may not have full network connectivity between all locations, which increases the implementation cost of such tools;
- Auditors may not have permission to install tools which require full system administrator privileges;
- The organisation may not authorise the implementation of such tools, considering the organisation's limited insight into the potential impact of such tools;

- Commercial solutions do not cater for all the risks and controls in the client's environment, resulting in inadequate functionality.

4.5.3.3 Specific vulnerability assessment tools

There are several security assessment tools that are used by the system, security administrators and auditors to assess the security of IT architecture components. These tools typically have two modes of operation, namely a non-intrusive scan mode and an attack mode, which may disrupt the operation of the system (Finnigan, 2016). These tools may be product- or component-specific. Examples of database-specific vulnerability assessment tools are AppDetectivePro, Scuba (Imperva), NGSSQuirreL for Oracle and AppSentry (Finnigan, 2016).

4.5.3.4 Password strength and hacking tools

Password strength tests can be executed to determine whether any password associated with an access path component is easy to guess (Davis *et al*., 2011). Free-of-charge and commercial password strength or hacking tools are easily available. Examples of such tools include AppSentry and NGSSQuirreL (ISACA, 2009). Without configured password controls, these tools can recover 90% of passwords in ±30 seconds (ISACA, 2009). This type of tool is however limited to password strength tests and serves a limited purpose in continuous auditing (ISACA, 2009).

4.5.4 Report and manage results

Similar to audit tool selection, planning for continuous auditing should include selecting reporting mechanisms (IIA, 2015). Successful continuous auditing programs support decision making and the remediation of any control deviations (IIA, 2015). A variety of reporting solutions may be implemented to meet the needs of management, the risk and compliance functions and the board of directors. This may include publishing electronic reports, as well as using traditional reporting formats, which include root-cause analysis and management's action plans (IIA, 2015).

The IIA (2015) recommends that reporting strategies are adapted to include sharing exceptions with management via electronic mechanisms in the following ways:

- Exceptions could be exported to a shared network folder or secure database for management review.
- E-mail notifications and workflow remediation tracking tools could be deployed to notify management in real-time of any exceptions.
- Trending information can be presented using web-based dash boards and data visualisation tools.

Deloitte (2016) recommends that interactive electronic reporting mechanisms are deployed to ensure that internal audit functions remain relevant in the modern business environment.

4.6 Conclusion

The implementation of continuous auditing procedures should follow a structured framework that includes planning at both strategic and operational levels, as shown in Figure 4.10.

Planning for continuous auditing commences at a strategic level when developing the audit universe and resulting annual audit plan. At this stage, a maturity assessment is conducted of existing continuous auditing initiatives to identify the business processes where the continuous auditing methodology could be implemented or improved (IIA, 2015). An overall implementation strategy is required to ensure support from management as well as the risk and compliance functions (IIA, 2015).

Each business process is then further analysed by means of a risk assessment to identify risk and control indicators, which is evaluated to determine whether it is suited to continuous auditing (IIA, 2015). Any (or all) of the four elements of continuous auditing (namely continuous data auditing, control monitoring, risk monitoring and compliance monitoring) could be relevant to the business process under review, depending on the risk assessment (Bumgarner & Vasarhelyi, 2015).

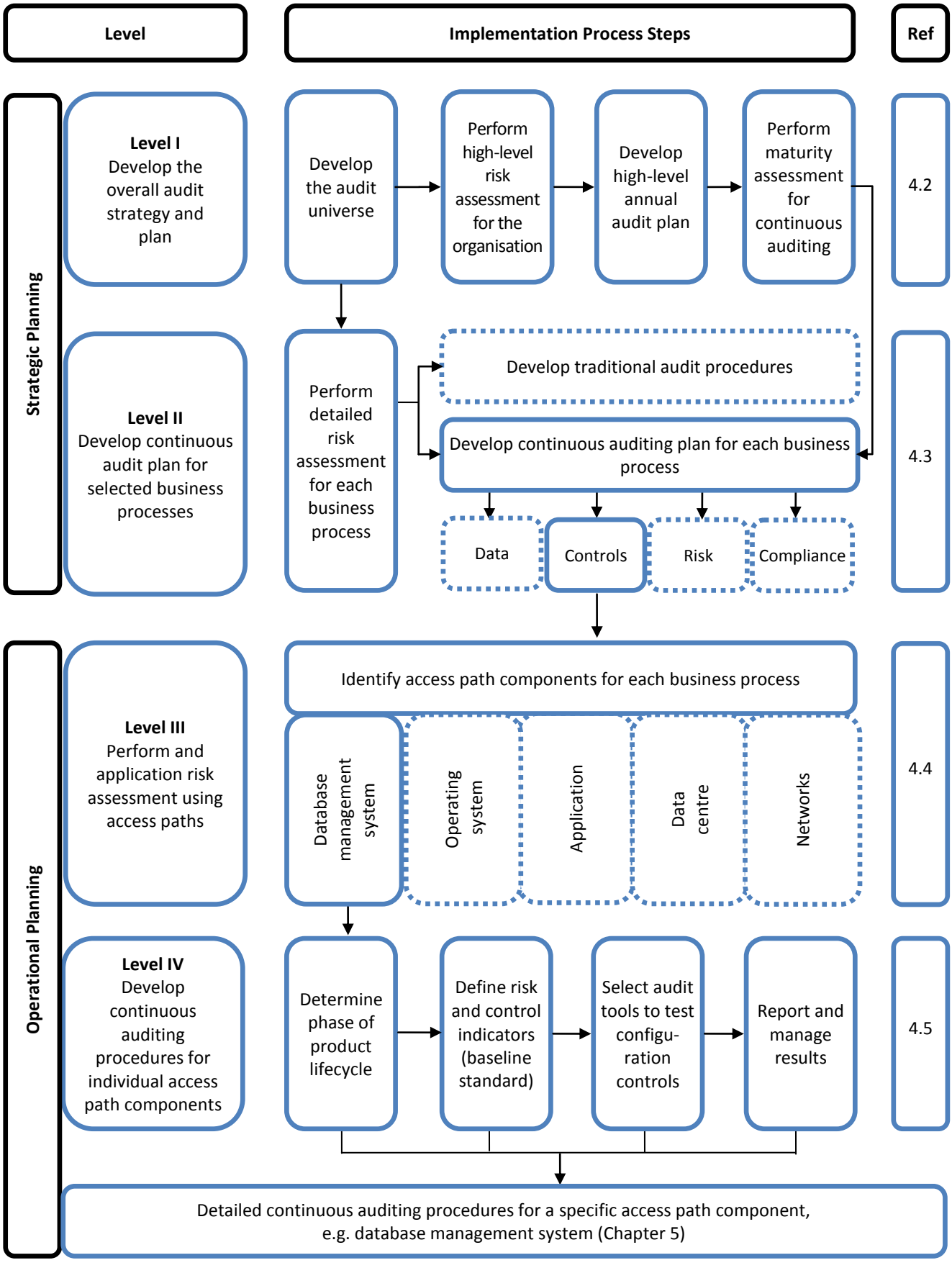
In this study, only one element of continuous auditing, namely **continuous control monitoring,** is examined at an operational level. Continuous control monitoring is the

ongoing assessment of configured (automated) controls (Bumgarner & Vasarhelyi, 2015). The IT access path model developed by Boshoff (1990) could be used to identify the relevant IT architecture components which could be audited using continuous audit techniques. In this manner, all possible access points, which are activated when an end-user accesses a computer-controlled resource, are identified (Goosen, 2012).

A detailed risk assessment is conducted for each access path component, considering the lifecycle phase of the particular component. Thereafter, a baseline standard is developed for the identified configuration controls for that particular access path component (IIA, 2015). Generalised audit software can be used to continuously compare the baseline standard with the configuration settings extracted directly from the system under review (Cooke, 2014). Any changes in these settings could be an early-warning indicator of a potential control violation or deficiency (IIA, 2015).

Using the audit planning framework developed in this study (refer to Figure 4.10), continuous auditing procedures (i.e. continuous controls monitoring) were subsequently developed for a widely-used database management system, namely Oracle Database, in Chapter 5.

Figure 4.10 Planning framework for developing continuous auditing procedures



(Source: Authors own construct)

CHAPTER 5. FINDINGS: DEVELOPING CONTINUOUS AUDITING PROCEDURES FOR ORACLE DATABASE MANAGEMENT SYSTEMS

5.1 Introduction

Continuous audit procedures relating to configuration controls entail the ongoing evaluation of automated controls against a baseline standard to identify configuration changes and risk indicators, as discussed in Chapter 4. The generic audit planning framework developed in Chapter 4 could be applied to develop continuous auditing procedures for any IT architecture component such as database management systems. In this chapter, the generic framework is applied to describe the continuous auditing procedures, specifically focusing on database management systems in each relevant phase of the product's lifecycle. The examples are limited to Oracle Database only.

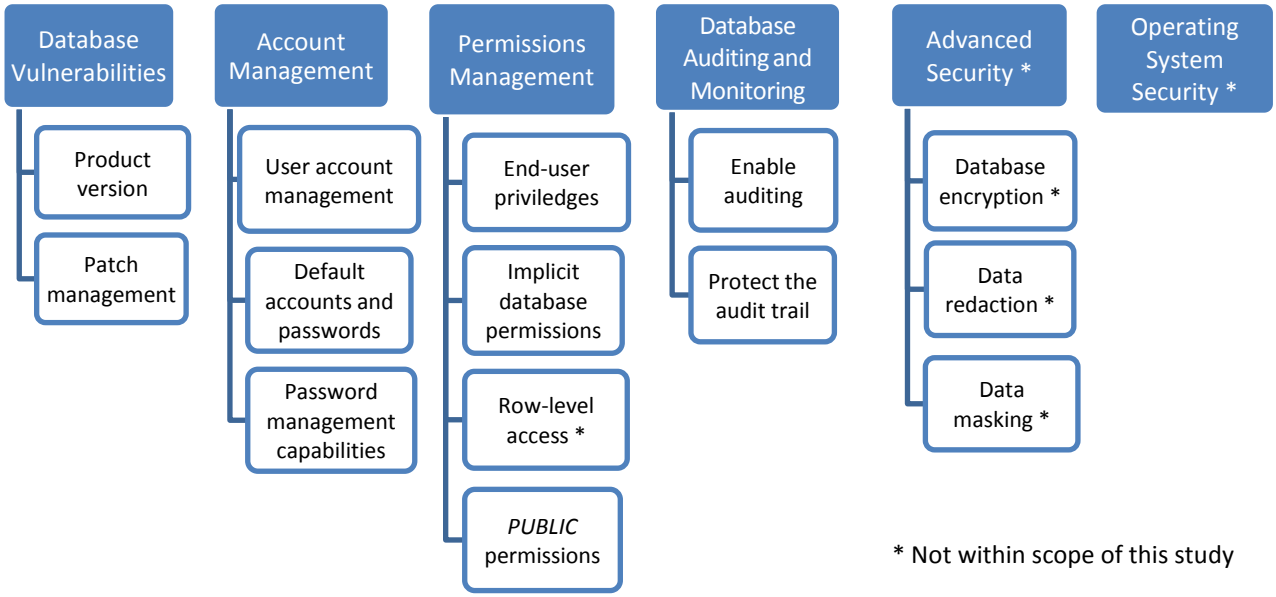
Sensitive and valuable organisational data (such as personal information, intellectual property and financial data) is typically stored in databases. Database management systems are therefore often the target of security breaches. McAfee (2015) noted that 68% of data breaches were of such significance that it required public disclosure or had a negative financial impact. Internal perpetrators (i.e. employees, contractors and suppliers) were responsible for 43% of the breaches, while external perpetrators were responsible for 57% of breaches (McAfee, 2015). Customer and employee information, including personally identifiable information, were in the top two content categories, followed by intellectual property and payment card information (McAfee, 2015). Therefore, internal controls should ensure that databases, which contain valuable organisational data, are protected against internal and external breaches. As such, the risks related to the validity, integrity and confidentiality of data is of a primary concern to the auditor (Davis *et al*., 2011).

The continuous auditing procedures described in this chapter are therefore limited to only those controls that ensure the validity, integrity and confidentiality of data, while the controls that ensure system availability are excluded.

5.2 Configurable controls for database management systems

Configuration controls for database management systems addressing the validity, integrity and confidentiality of data can typically be grouped into six control categories, as shown in Figure 5.1.

Figure 5.1 Control categories for database management systems



(Sources: Rahman, 2014; Cooke, 2014; Davis *et al*., 2011; ISACA, 2009)

- **Database vulnerabilities** could be exploited by unauthorised parties to gain access to sensitive data within the database. Therefore, database management systems should be operated using supported versions of the software (Rahman, 2014). Patch updates, i.e. repairs to software programming errors and vulnerabilities (ISACA, 2016), should also be current (Rahman, 2014). Refer to paragraph 5.3.

- **Account and password management** includes the management of user accounts, account creation and termination, naming conventions and password management capabilities. Refer to paragraph 5.4.

- **Permissions management** entails assigning database privileges to users on a least-privilege principle (Rahman, 2014). In particular, administrators and end-users should only have access to the minimum roles and privileges which are required to perform their job function (Rahman, 2014). Refer to paragraph 5.5.

- **Database auditing and monitoring** are implemented to create an audit trail of selected high-risk activities on the database, by both database and non-database users (Miller & Kost, 2016). This audit trail is not only required in regulated industries, but may help to ensure accountability of users by tracking their actions (Huey, 2016). Refer to paragraph 5.6.

- **Advance database security measures** such as data encryption, data masking (changed or substituted data) and data redaction (scrubbed or hidden data) could be introduced in environments that require additional controls for high-risk data elements (Huey, 2014). Advanced security measures are excluded from the scope of this research as these measures are typically only implemented in highly controlled environments and are often implemented using proprietary software in addition to the database management system software.

- **Operating system security** – Insufficient security of the operating system layer of the access path may expose the database management system environment to security risks (Davis *et al.*, 2011). The operating system is a separate access path component and is excluded from the scope of this research which focuses on database management systems only.

Cooke (2014) proposes an audit approach to extract and review the above control categories using generalised audit software in conjunction with information taken directly from the particular database management system under review, such as the data dictionary and certain initialisation parameters.

The data dictionary contains information about every object in the database, including users, privileges, roles and auditing information. The contents of the data dictionary can be queried using the Oracle Database data dictionary views and initialisation parameters which can be retrieved from the *V$PARAMETER* view (Cooke, 2014). There are approximately 260 Oracle Database initialisation parameters (Cooke, 2014). Examples of parameters include those for enabling the auditing of the activities of privileged users and security settings such as password parameters (Cooke, 2014). Not all the parameters of interest need to be reviewed for all databases. The master profile should only include

those parameters mitigating the risks relevant to the particular database, i.e. the configuration control categories listed in Figure 5.1 (Cooke, 2014).

Data dictionary views are static and are prefixed with '*DBA_'*. Similarly, Oracle Database maintains dynamic performance views that are continuously updated. These dynamic views are prefixed with *'V$'* (Cooke, 2014). All static data dictionary (*DBA_*) and dynamic (*V$*) views, together with detailed explanations, can be found in the Oracle Database Reference which is published on the Oracle website (Rich, 2016).

Cooke (2014) suggests that the static dictionary (*DBA_*) and dynamic (*V$*) views are extracted for auditing using the SQL*Plus query tool. SQL*Plus is included in every Oracle database installation and enables querying the database using Structured Query Language (SQL) (Cooke, 2014). The output can be formatted as required to be directed to a file (e.g. comma-separated value (CSV) text file) for further analysis using generalised audit software (Cooke, 2014).

Using generalised audit software, the extracts are compared to baseline standards in the configuration phase of the product's lifecycle (IIA, 2015). Thereafter, the extracts are analysed for any changes since the previous extract, as described in paragraph 4.5.2.

The continuous audit procedures listed in this chapter are addressed for the four control categories within the scope of this research, as depicted in Figure 5.1. For each control category, the risks, controls and audit procedures are discussed briefly. Both traditional and continuous auditing procedures are described using Oracle Database as example. The continuous auditing procedures are subsequently tabled for each component, including the SQL query to extract the relevant audit data. The tables also indicate the phases of the product lifecycle which are relevant for each procedure. Only the configure, operate and maintain phases are included in the tables, since the build phase is not relevant for database management systems, as this phase pertains mostly to hardware components. In addition, the set-up phase is excluded, as the risks and controls are similar to those of the configure phase.

5.3 Database vulnerabilities: Product version and patch management

Database vulnerabilities may be exploited by internal and external perpetrators to gain unauthorised access to the database (McAfee, 2015). This risk is mitigated by ensuring the database management system's product version remains supported by the product vendor and that security patches are up-to-date (Cooke, 2014).

5.3.1 Product version

The product version of legacy databases may no longer be supported by the database vendor (Davis *et al*., 2011). For example, support for Oracle Database 11g Release 2 was suspended in January 2015 and extended support will end in December 2020 (Oracle, 2016).

Risks

The increasing risk of unsupported software could result in software no longer receiving security updates, software (program) code updates or online technical content updates (Davis *et al.,* 2011). Compared to legacy systems, modern systems are protected by advanced security technologies which are specifically designed to increase the complexity of the attack to be carried out by cyber criminals in order to exploit any particular vulnerability (Microsoft, 2013). The appeal for cyber criminals to exploit vulnerabilities is therefore reduced for modern product versions (Microsoft, 2013).

Controls

An organisation's IT standards and policies should define the requirements and processes to ensure that all implemented database product versions are supported by the particular software vendor (Davis *et al.,* 2011).

Procedures

Traditional audit procedures entail determining which product versions are recommended and implemented by the organisation. Thereafter, a register of the relevant databases and the respective versions should be obtained or compiled by the auditor (Davis *et al*., 2011). The final step is to verify whether the product versions in use by the organisation are still supported by the software vendor (Davis *et al*., 2011). The auditor can determine the product version by browsing system management modules or the data

dictionary (Microsoft, 2016), while software vendors publish details of supported product versions on the internet (Davis *et al*., 2011).

Continuous audit procedures entail an initial policy review, similar to the traditional audit procedures. However, procedures to determine the product version are automated using scripts. Audit procedures to test the version include the following:

- Although it is not expected that unsupported product versions will be installed in the configuration phase, testing may be conducted to determine the version, in order to aid the risk assessment and the development of other procedures (refer to Table 5.1 item 1).
- The product version extracted using the query statement can be compared to supported versions as published by the software vendor using analytical software (refer to Table 5.1 item 1).

5.3.2 Patch management

Software vendors publish regular scheduled software updates, known as patch releases, to maintain up-to-date software, repair software programming defects and address security vulnerabilities (Hoehl, 2013). Patch management is an area of systems management that involves acquiring, testing and installing multiple patches (code changes) to computer systems (ISACA, 2016).

Risks

With the proliferation of malware and network intrusions, it has become more critical to implement patch updates in a timely manner in order to protect information systems (Hoehl, 2013). Not only may unpatched systems render systems vulnerable to malicious attacks (causing system downtime and unauthorised disclosure of confidential data, for example), but it may also impair the organisation's ability to conduct business (Hoehl, 2013). For example, credit card associations such as American Express may impose non-validation fees on merchants and may terminate the agreement if merchants do not fulfil contractual requirements regarding patch updates; that is, merchants may not continue to accept credit cards if patch updates are not implemented timeously (Hoehl, 2013). In addition, customers may rather prefer to do business with organisations that do comply with security best practices (Hoehl, 2013).

Although the application of patch updates is intended to mitigate risk, this process is potentially disruptive and may introduce significant risks if it is not managed appropriately (IIA, 2012). Patch updates typically affect critical system libraries and other software used by database management systems. Although patch updates tend to entail significant changes, there is often limited documentation available that describes the underlying changes (IIA, 2012). As a result, small configuration variances may have unexpected and disruptive results; for instance critical systems, including databases, may be unavailable for a prolonged period (IIA, 2012).

Controls

Organisations should have policies and procedures to identify and timeously apply any new patch updates. Rahman (2014) recommends that database administrators subscribe to the particular software vendor's notification service to be alerted of scheduled and unscheduled patch updates. For example, Oracle patches and security alerts are published via the Oracle e-mail security alert advisory service (Rahman, 2014).

Patch management should be considered as a subset of change management, and implementing patch updates should follow the same process as for any other change implemented in the IT environment (ISACA, 2016; IIA, 2012). Critical patch updates can have a significant impact on the database, depending on the database schema. Therefore, extensive regression testing may be required to ensure that applying the latest patch update has no impact on the database functionality (Rahman, 2014).

Procedures

Davis *et al*. (2011) propose a **traditional audit methodology** that entails interviewing the database administrator and reviewing the policies and procedures which ensure that patch updates are identified, tested and applied systematically and timeously. The auditor should also determine how the risk related to each patch is assessed and whether alternative mitigating controls are considered instead of installing the patch update, such as removing the system components that introduce the particular vulnerability or blocking the vulnerability with a firewall (IIA, 2012).

Following the policy review, a register of the databases together with its patch should be obtained from the database administrator or compiled by the auditor (Davis *et al.*, 2011).

The final step is to determine the latest patch update details published by the software vendor (Davis *et al*., 2011).

Continuous audit procedures for patch management entail an initial policy and process review, similar to the traditional audit procedures. The same queries as for the product version can be utilised to determine the patch status of a database (refer to paragraph 5.3.1).

Audit procedures to test the patch status of the database include the following:

- Although is not expected that unpatched software will be installed in the configuration phase, testing may be conducted to determine the patch levels (refer to Table 5.1 item 2).
- The patch level extracted using the query statement can be compared to the latest patch levels published by the software vendor using analytical software (refer to Table 5.1 item 2).

Table 5.1 Continuous audit procedures: Database vulnerabilities

Risk/Control Area		SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
			Configure	Operate	Maintain
1.	**Product version**	Query statement to extract product version: *SELECT* FROM V$VERSION* Compare to product version information published online (typically available in table format for further analysis).	**MAYBE ****	**NO**	**YES**
2.	**Patch management**	Query statement to extract patch level: *SELECT* FROM V$VERSION* Compare to patch information published online (typically available in table format for further analysis).	**MAYBE ****	**NO**	**YES**
			** Procedure may not necessarily be relevant, as it is not expected that unsupported or unpatched software will be installed in the configuration phase.		

(Sources: Cooke, 2014; Rahman, 2014; ISACA, 2009)

5.4 Account and password management

User account and password management are the primary controls to restrict access to the database (Davis *et al*., 2011). User account management, username and password conventions as well as password parameters are the control areas that are relevant to ensure access is restricted to authorised users only (Rahman, 2014).

5.4.1 User account management

Risks

In the absence of appropriate user account management procedures, users may obtain inappropriate or unauthorised access to the database (ISACA, 2009). Risks include unauthorised changes to data and the exposure of confidential, sensitive or regulated information (Gibbs *et al.,* 2010). In addition, if generic (shared) user accounts are used, individual users cannot be associated with specific actions performed on the database and can therefore not be held accountable for those actions (ISACA, 2009).

Controls

Application end-users should typically not have direct access to a database, but should rather gain access to the database through the application front-end (Davis *et al*., 2011). Therefore, to limit unnecessary access, organisations should also implement effective controls for the provisioning and revocation of access to databases. This includes the procedures for creating user accounts and ensuring that accounts are created only for legitimate business requirements, particularly when privileged access is to be granted (Davis *et al*., 2011). Also, the organisation should implement processes to identify and revoke user accounts timeously following the termination of employment or a change in the job requirements of the particular use. This may include a periodic access review by the database administrator (Davis *et al*., 2011).

Procedures

Traditional audit procedures include reviewing the user list for application end-users with database access for any instances of inappropriate access. As direct database access is not desirable, the auditor should review the need for any users to have such access. In addition, no guest or generic (shared) accounts should exist (Davis *et al*., 2011). A sample of new users could be reviewed to establish whether the standard authorisation process was followed (Davis *et al*., 2011). The access granted can also be reviewed in relation to

the particular user's job function. The account termination process could be confirmed by reviewing a sample of users to identify terminated users and those users whose job functions have changed (Davis *et al*., 2011).

The **continuous audit process** will commence in the configuration phase with extracting the database users, together with their roles and privileges, using database queries and analytical software (Cooke, 2014). A user list containing the user names, account status, last login dates, roles and privileges is not readily available, but can be constructed using analytical software by joining the various tables listed in Table 5.2 (ISACA, 2009).

Table 5.2 Oracle DBA tables for user access review

DBA view	Description
DBA_USERS	User accounts with status (open, locked, expired) and last login date
DBA_ROLES	Defined roles and authentication type (i.e. password protected)
DBA_ROLE_PRIVS	Mapping of users to assigned roles
DBA_SYS_PRIVS	System privileges granted to users and roles
DBA_TAB_PRIVS	Object privileges granted to users and roles
ROLE_ROLE_PRIVS	Roles granted to other roles
ROLE_SYS_PRIVS	System privileges granted to each role
ROLE_TAB_PRIVS	Table privileges granted to each role

(Sources: Cooke, 2014; ISACA,.2009)

- The initial user list is reviewed for appropriate access, similar to the traditional review. Using generalised audit software, this user list is then compared to an extract of current employees and their job descriptions as per the human resources system (Cooke, 2014). This comparison should also highlight potential generic accounts (Davis *et al*., 2011). Once validated in the configuration phase, the initial user list is used as baseline to identify any new users added since the last review or any users with changed job descriptions (Cooke, 2014). This is repeated periodically during the maintain phase of the product's lifecycle. Refer to Table 5.5 item 1.

- Dormant accounts can also be identified by reviewing the last date that the particular user logged onto the database, using the same extracts as above (Miller & Kost, 2016). Refer to Table 5.5 item 2.

5.4.2 Default accounts and passwords

Risks

Default database accounts with default usernames and passwords are created when the database is implemented, or later during the operation of the database, for example when performing upgrades (Finnigan, 2016). As default accounts and password are well-known and widely published, an attacker (e.g. an external hacker, internal employee or malware) can exploit a default account to gain unauthorised access to the database (Finnigan, 2016). As default accounts typically have critical system privileges, unauthorised parties could gain access to both view and change sensitive data (Finnigan, 2016). Widely available hacking tools such as Metaspoilt and Backtrack have a high success rate when attempting to compromise default accounts with default passwords (ISACA, 2009).

Details of default database accounts of widely used systems are listed on various security websites (Rahman, 2014). For example, ±600 default Oracle accounts and passwords (refer to Table 5.3) are published online (Finnigan, 2016).

Table 5.3 Examples of Oracle default accounts

Default Account	Default Password	Description
SYSTEM	MANAGER	Database management account with privileges to read, change and delete data in the database
SYS	CHANGE_ ON_INSTALL	The most powerful database management account with privileges to read, change and delete data
DBSNMP	DBSNMP	Under certain circumstances, DBSNMP allows to read passwords from memory
SYSMAN	SYSMAN	The management account for Oracle Enterprise Manager which is used to access all databases that are managed by it, and may access all data in these databases

(Sources: Finnigan, 2016; Rahman, 2014)

The risk is reduced for new installations of later versions of Oracle Database (i.e. 11g and 12c) for which the default accounts are either locked or require a new password during the

database installation (Finnigan, 2016). This does however not address the risk for Oracle installations which were upgraded from previous versions (Finnigan, 2016).

Controls

Database administrators should ensure that all default accounts are configured with a strong password. If a particular account is not required, the account should be locked and expired (Rahman, 2014). For later versions of Oracle Database (i.e. 11g and later), the Oracle database configuration assistant (DBCA) automatically locks and expires the majority of the default database user accounts, unless the database is installed manually (Rahman, 2014). The DBCA also changes the password of the *SYSTEM* account to the value specified during the installation routine (Rahman, 2014).

Procedures

Traditional audit procedures entail reviewing the user list for default accounts and then attempting to log onto the database using the default passwords (ISACA, 2009).

Continuous auditing procedures involve the automation of the above-mentioned traditional audit procedure to identify all default accounts and passwords, by extracting the user tables and comparing the password hashes with pre-computed password hashes published on the Internet (Finnigan, 2016). Refer to Table 5.5 item 3.

Audit procedures to identify default accounts and passwords include the following:

- In the configuration phase, default accounts are created when the database is installed. For example, the *SYS* and *SYSTEM* accounts are created by default when the database is installed by using a wizard (Finnigan, 2016). Continuous auditing procedures should be developed to identify any such default accounts for remediation by management. No such accounts should have default passwords, even though the system is still in the configuration phase (Cooke, 2014).
- The process is repeated during the operate and maintain phases to identify any default accounts which were installed or unlocked subsequent to the initial configuration. Further default accounts can be created after the initial database installation when executing default scripts, located in the *$ORACLE_HOME/ rdbms/admin* directory, to add an additional feature to the database (Finnigan, 2016). In addition, default database users are also created when third party software (e.g. SAP) is installed (Finnigan, 2016).

If default accounts and passwords are limited as per the best practice guidelines, the continuous auditing procedures should deliver very little exceptions (Cooke, 2014).

5.4.3 Password management capabilities

Risks

Users often choose passwords that can be guessed easily by automated programs (Davis *et al.,* 2011). Unless password management capabilities are enabled, weak passwords could be exploited to gain unauthorised access to the database (Davis *et al.,* 2011). In the absence of password management features, these tools can recover 90% of passwords in less than 30 seconds (ISACA, 2009).

Although modern database management systems may have robust password management capabilities, the features are not necessarily enabled (Rahman, 2014). For example, Oracle password parameters are assigned to profiles which, in turn, are assigned to users (Rahman, 2014). The default Oracle profile (with unlimited parameters) is assigned to all users where no alternative profile is specified when the user is created (Rahman, 2014).

Earlier versions of Oracle Database (i.e. version 8 and earlier) do not have any password management capabilities (ISACA, 2009). From Oracle Database 11g onwards, enhanced password controls such as password strength validation and case-sensitive passwords are available, but may still be disabled (ISACA, 2009).

Controls

Database management systems typically include rich password management features. Oracle includes features for password expiry, re-use limits, lockout and lockout reset (Rahman, 2014). Refer to Table 5.4 for a summary of Oracle Database password parameters and the recommended benchmark setting for Oracle 12c.

Table 5.4 Description of recommended Oracle 12c password parameters

Password Parameter	Description	Benchmark
FAILED_LOGIN_ATTEMPTS	The maximum number of failed login attempts before an account is locked	5 attempts
PASSWORD_LOCK_TIME	The number of days that an account will be locked if the maximum number of failed login attempts is exceeded	1 day
PASSWORD_LIFE_TIME	Password expiration, i.e. the number of days before a particular password expires	90 days
PASSWORD_GRACE_TIME	The grace period before a password expires, i.e. following a warning message	5 days
PASSWORD_REUSE_TIME *	Password history, i.e. the number of days before a previously used password can be reused	90 days
PASSWORD_REUSE_MAX *	Password history, i.e. the number of times that a password must be changed before a previously used password can be reused	10 times
PASSWORD_VERIFY_FUNCTION	Evaluates the complexity of a password, e.g. test for password length, special characters, dictionary words, username, etc.	Enabled
SEC_MAX_FAILED_LOGIN_ATTEMPTS	Restricts the number of failed authentication attempts from a particular connection to slow down brute force (hacking) attacks	10 times
SEC_CASE_SENSITIVE_LOGON	Passwords are case sensitive by default only for Oracle 11g and later	Enabled
* Password reuse_time and reuse_max are mutually exclusive.		

(Sources: CIS, 2015; Rahman, 2014)

Procedures

Traditional audit procedures for password management commence with an interview with the database administrator to ascertain whether the password parameters available for the database system have been enabled (Davis *et al*., 2011). Once the configuration values for password management have been obtained, the auditor should ensure that each feature is enabled according to the organisation's information security policies (Davis *et al*., 2011). The auditor should also evaluate whether a particular setting is appropriate, considering the particular risk assessment for that organisation and system (Davis *et al*., 2011).

Continuous audit procedures for password management entail the automation of the traditional audit procedures, by reviewing the password configuration settings and testing for default profiles (Rahman, 2014).

Audit procedures relating to password management capabilities include the following:

- The password management parameters are extracted from the applicable Oracle table for comparison to best practice or organisational security standards during the configuration phase (ISACA, 2009). This test is repeated periodically in the maintain phase to detect any changes to the standard password parameters. Refer to Table 5.5 item 4.
- Procedures should be included in the configure and maintain phases to identify any user accounts that have been assigned the default password profile (i.e. with unlimited password parameters) (ISACA, 2009). No user should have this profile (ISACA, 2009). Refer to Table 5.5 item 5.
- In addition to the continuous configuration procedures, password strength tests can be executed on password hashes to determine whether any passwords are easy to guess (Davis *et al.*, 2011). Password strength tools are widely available, both free and commercially, as discussed in paragraph 4.5.3.4. Refer to Table 5.5 item 6.

Table 5.5 Continuous audit procedures: User account and password management

Risk/Control Area		SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
			Configure	Operate	Maintain
1.	**Direct end-user access**	Direct database users can be identified by querying the *SYS.DBA_USERS* table: *SELECT USERNAME, ACCOUNT_STATUS*	**YES**	**NO**	**YES**
2.	**Dormant users**	Dormant users can be identified by the following query of the *SYS.DBA_USERS* table: *SELECT USERNAME, ACCOUNT_ STATUS, COMMON, LAST_LOGIN FROM SYS.DBA_USERS* Dormant users can be identified by interrogating the date information in the *LAST_LOGIN* column of the table.	**NO**	**NO**	**YES**
3.	**Default usernames and passwords**	Default user names and passwords are identified by extracting *DBA_USERS* and matching the password hashes to pre-computed password hashes published on the internet: The database view *DBA_USERS_WITH_ DEFPWD* (11g and later) lists all default accounts with default passwords.	**YES**	**YES**	**YES**

Risk/Control Area		SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
			Configure	Operate	Maintain
4.	**Password management: Password parameters**	Password parameters are extracted from *DBA_PROFILES* and compared to standards as per Table 5.4. *SELECT PROFILE, RESOURCE_NAME, LIMIT FROM DBA_PROFILES WHERE RESOURCE_NAME="X"* X = Parameters as per Table 5.4	**YES**	**NO**	**YES**
5.	**Password management: Default profiles**	User accounts with default profiles can be extracted from the *DBA_USERS* view using the following script: *SELECT USERNAME FROM DBA_USERS WHERE PROFILE = 'DEFAULT''* No user should have this profile.	**YES**	**NO**	**YES**
6.	**Password management: Password strength**	Extract *DBA_USERS* for further analysis using password strength test tools such as John the Ripper, Oracle Auditing Tools (OAT) or NGSSQuirreL	**YES**	**NO**	**YES**

(Sources: Miller & Kost, 2016; Cooke, 2014; ISACA, 2009)

5.5 Database permissions management

5.5.1 Database permissions – background

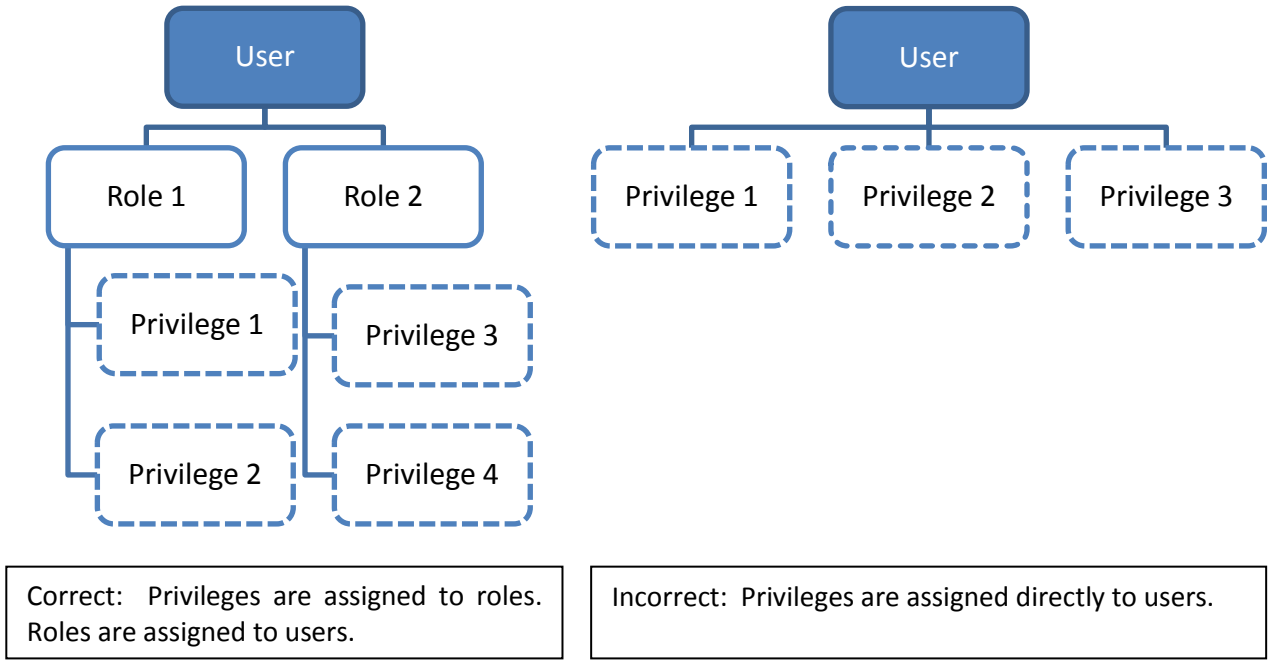
Permissions management entails assigning database privileges to users on a least-privilege principle; that is, administrators and end-users should only have access to the minimum roles and privileges that are required to perform their job function (Rahman, 2014). Continuous audit procedures could be utilised to identify any administrators and users with either excessive or unauthorised access (CIS, 2015).

Database privileges are managed in two categories, namely system privileges and object privileges ISACA, 2009). System privileges allow the user to create or manipulate objects such as tables or triggers, e.g. *CREATE TABLE, CREATE ANY TRIGGER, DROP ANY PROCEDURE* (ISACA, 2009). A database object is anything that exists in a database and on which operations can be performed (e.g. program files, triggers, libraries, tables, views and folders) (ISACA, 2009). Object privileges (*SELECT, INSERT, DELETE*, etc.) enable the user to access data with an object. For example, any user with *DELETE* access to the *ORDERS* table will be able to delete a row from the table containing order information (ISACA, 2009).

Both object and system privileges introduce the risk of unauthorised access or changes to data; therefore, this type of access should preferably be restricted to database administrators (ISACA, 2009). Audit procedures during both the configure and maintain phases of the product's lifecycle should identify any object and system privileges assigned to user accounts which do not have a database administrator role (Rahman, 2014)

Although privileges could be assigned directly to users, these privileges should preferably be assigned to roles (groups) which should then in turn be assigned to users (Larner, 2014). This allows for easier maintenance of security controls and reduces the risk of administrative mistakes when assigning roles (ISACA, 2009). The correct manner to assign privileges to roles is depicted in Figure 5.2.

Figure 5.2 Role versus privilege assignment for user accounts



(Source: Author's own construct)

5.5.2 Review database privileges granted to end-users

Risks

Application end-users should typically not have direct access to a database, but should rather gain access to the database through the application. Should direct access be granted to database users, there is an increased risk of inappropriate access to change or view critical data (Rahman, 2014).

In addition, database users may be assigned access to data and privileges which are not required in terms of their job function (Davis *et al*., 2011). The risk of excessive privileges increases when privileges are assigned directly to users, instead of indirectly through the use of roles (also referred to as user group membership) (ISACA, 2009).

Controls

Should direct access be necessary, Rahman (2014) recommends that organisations practise the principle of least privilege; that is, database users (or roles) should not be granted more privileges than necessary in order to perform their job function. Database security standards should specify those systems and object privileges that should not be granted to end-users (Davis *et al*., 2011)

Database best practice recommends that permissions are granted to roles (user groups) instead of individuals (Davis *et al*., 2011). Roles or group membership is then assigned to the individuals. This principle should be included in the documented database standards (ISACA, 2009).

To implement the principle of least privilege, Rahman (2014) recommends that high-risk permissions should either not be assigned to end-users, but rather to privileged users only. In particular:

- End-users should not have the ability to create, modify or delete database objects via permissions such as *TRUNCATE TABLE, DELETE TABLE, DROP TABLE*.
- The role required to export the full database (*EXP_FULL_ DATABASE*) could enable the extraction of data for unauthorised distribution.
- Library-related privileges (*CREATE LIBRARY* and *CREATE ANY LIBRARY*) enable the creation of library objects and could corrupt the library's integrity.

Procedures

Audit procedures should be developed to ensure that database access is limited to users with legitimate reasons for direct database access. **Traditional audit procedures** entail a discussion with the database administrator to determine which end-user accounts are required to have direct access to the database, and to what data (Davis *et al.,* 2011). End-users should be distinguished from administrator accounts and system accounts (e.g. web application accounts) and accounts used to process batch jobs (Davis *et al.,* 2011).

The traditional audit also includes a manual review of the database dictionary for permissions that were granted to an account or user, instead of using roles or groups. This is done with the assistance of the database administrator (ISACA, 2009).

For **continuous auditing** purposes, the traditional audit procedures are automated, instead of relying on the database administrator for this information. Audit procedures to test the database privileges granted to end-users include the following:

- The auditor compiles a user list with role and privilege details by joining the tables containing usernames, roles and privileges (ISACA, 2009). During the configuration

phase, the audit objective is to determine whether the access associated with each user is appropriate considering their job function (Rahman, 2014). The procedure is then repeated continuously during the maintain phase to identify any new users with excessive database privileges. Refer to Table 5.6 item 1.

- During both the configure and maintain phases, any end-users with privileged roles, such as database administrator, should be identified (Huey, 2016). Typically, no end-user should have this role (Huey, 2016). Refer to Table 5.6 item 2.
- No user should have any permissions which were granted directly to the user account instead of indirectly via a role, as demonstrated in Figure 5.2 (ISACA, 2009). The assignment of privileges directly to roles is identified by reviewing the above user list that is compiled using generalised audit software (ISACA, 2009). No exceptions should be noted if the recommended practice is followed. Refer to Table 5.6 item 3.

5.5.3 Implicit database permissions

Risks

Database permissions are not always necessarily assigned explicitly to end-users and roles (ISACA, 2009). Instead, unintended access and privileges could be assigned through implicit privileges (ISACA, 2009). Such implicit privileges could also result in unauthorised access to sensitive data and unauthorised access to delete data or critical tables (Davis *et al.,* 2011).

For Oracle Database, implicit privileges typically result from substitute (*ANY* and *BECOME*) privileges that are assigned to users (ISACA, 2009). Examples include the following:

- The *SELECT ANY TABLE* privilege allows the particular user to read any table in the database, except the *SYS* table. This privilege should be limited to the database administrator (Huey, 2016).
- *DROP ANY TABLE* allows the users to delete any table in the database (Rahman, 2014).
- The *BECOME USER* privilege allows the designated user to inherit the privileges of another user (CIS, 2015).

Likewise, user administration rights can also be implicitly granted to non-administrative users (ISACA, 2009). For example:

- *GRANT ANY OBJECT PRIVILEGE/ROLE/PRIVILEGE* enables the grantee to assign all objects, privileges or roles to another user (Huey, 2016).
- Limited user administrator rights are assigned to users with privileges such as *WITH GRANT OPTION* and *WITH ADMIN OPTION* (ISACA, 2009). This effectively gives the user administrator rights to a particular object or system privilege (i.e. the user is able to re-assign that particular privilege to another user) (ISACA, 2009).

Controls

The principle of least privilege should be applied when assigning permissions that could result in unintended system-wide access or user administration rights (Rahman, 2014). The documented database standards should describe those high risk permissions that should be limited to avoid implicit (substitute and administrator) access that is not required (Davis *et al*., 2011). Software vendors and standard setters, such as the CIS, publish potential implicit access for particular databases (CIS, 2015).

Procedures

Implicit database permissions cannot be reviewed effectively by using manual **traditional audit procedures**. **Continuous auditing procedures** commence with the traditional policy and documented standards review, and implicit database permissions are identified by reviewing extracts of permission tables (ISACA, 2009). Extracts are designed to identify any users with substitute permissions (refer to Table 5.6 item 4) and implicit administrator rights (refer to Table 5.6 item 5) (CIS, 2015). There should typically be no users with such privileges (CIS, 2015).

5.5.4 Row-level access to table data

Risks

Relational databases are designed to grant permissions to users relating to a database table or columns, but are typically not designed to restrict access to a subset of rows in a table (Davis *et al.,* 2011). For example, when granted *SELECT* privileges in a table, the user will be able to read every row in that table. This could result in unintended wide access to view data (ISACA, 2009).

Controls

Several technologies can be used to address this problem. For example, Oracle offers virtual private databases (VPDs) which limit access to specific rows. Another common approach is to use stored procedures to access data; that is, users have access to the stored procedure instead of permissions to read the table (Davis *et al.*, 2011). A stored procedure is a collection of SQL statements that perform a particular task or set operations to be executed on a database server (ISACA, 2009). Similar to stored procedures, database views can be used to create a tailored view of a subset of data within a table or combination of tables (Davis *et al.,* 2011). The controls should ensure that the user cannot access the data in the table, should the user attempt to circumvent the database view or stored procedures (Davis *et al.*, 2011).

Procedures

Davis *et al*. (2011) proposes that **traditional audit procedures** commence with a discussion with the database administrator to determine whether row-level access is applicable, as well as the method of row-level access controls in the database. Thereafter, the auditor should access the database using a similar user account to verify the effective ability of that user account (Davis *et al*., 2011). As row-level access controls will vary among organisations and database instances, depending on the purpose of the database and the software application related to the particular database, detailed **continuous auditing procedures** for row-level access have been excluded from this study.

5.5.5 *PUBLIC* permissions

Risks

The *PUBLIC* group is installed by default for most databases and introduces security risks if the privileges assigned to this group are not restricted (Rahman, 2014). Privileges granted to the *PUBLIC* group are typically inherited by all accounts with access to that database (Davis *et al.*, 2011). Many of the built-in procedures and functions in a database are granted to the *PUBLIC* group by default (ISACA, 2009). As *PUBLIC* privileges are inherited by all other accounts, the *PUBLIC* role is not highlighted in the *DBA_ROLES* view as listed in Table 5.5 (Huey, 2016).

Controls

The database administrator should limit the privileges that are assigned to the *PUBLIC* role (ISACA, 2009). However, the blind revocation of permissions from the *PUBLIC* group is not recommended, as these permissions may be required to perform certain functionalities (Davis *et al*., 2011).

Procedures

Except for the review of policies and standards, manual **traditional procedures** are ineffective in reviewing *PUBLIC* privileges.

Continuous auditing procedures should include the extraction and review all privileges assigned to the *PUBLIC* role, including system and object privileges (Davis *et al*., 2011). This should be done during the system configuration phase. Any subsequent privilege additions to the *PUBLIC* role should also be detected and reviewed for authorisation in terms of the organisation's change control process (ISACA, 2009). During the configuration phase, the focus is to determine whether all privileges assigned to the *PUBLIC* role are required. Thereafter, during the maintain phase, the focus is to identify any new privileges assigned to the *PUBLIC* role and to confirm whether the additional roles are required and have been authorised (CIS, 2015). Refer to Table 5.6 item 6.

Table 5.6 Continuous audit procedures: Permissions management

Risk/Control Area		SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
			Configure	Operate	Maintain
1.	**End-user privileges**	Join the four DBA views below to compile user list with roles and privileges: • *DBA_SYS_PRIVS* • *ROLE_PRIVS* • *ROLE_ROLE_PRIVS* • *ROLE_TAB_PRIVS* Join the above user list with human resources data to associate users with employees and their job function. Using the above views, identify any roles and/or user accounts with the following privileges: • *INSERT* • *UPDATE* • *DELETE* • *TRUNCATE* • *DROP*	**YES**	**NO**	**YES**
2.	**End-user privileged roles**	Execute the following query to identify users with privileged roles such as the database administrator role: *SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTED_ROLE='DBA' AND GRANTEE NOT IN ('SYS','SYSTEM')*	**YES**	**NO**	**YES**

Risk/Control Area	SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
		Configure	Operate	Maintain
3. Direct end-user access	The following DBA views should be analysed to identify privileges assigned directly to end-users: • *DBA_SYS_PRIVS* • *DBA_TAB_PRIVS* • *USER_SYS_PRIVS* User accounts are listed in: • *DBA_USERS*	**YES**	**NO**	**YES**
4. Implicit permissions: Wide access	The following query statements could be used to identify users or roles with implicit or substitute privileges: *SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS WHERE PRIVILEGE = 'SELECT ANY TABLE'* *SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS WHERE PRIVILEGE = 'GRANT ANY X' AND GRANTEE NOT IN ('DBA', 'MDSYS', 'SYS', 'IMP_FULL_ DATABASE', 'EXP_FULL_ DATABASE', 'DATAPUMP_IMP_FULL_ DATABASE', 'WMSYS', 'SYSTEM','OLAP_DBA', 'DV_ REALM_OWNER') X= OBJECT PRIVILEGE, ROLE PRIVILEGE* *SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS WHERE PRIVILEGE= 'BECOME USER' AND GRANTEE NOT IN ('DBA','SYS', 'IMP_FULL_DATABASE')*	**YES**	**NO**	**YES**

Risk/Control Area		SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
			Configure	Operate	Maintain
5.	**Implicit permissions: Administration rights**	The following query statements could be used to identify users or roles with *ADMIN* or *GRANT* privileges: • *SELECT*FROM DBA_ROLE_PRIVS WHERE ADMIN_ OPTION = 'YES'* • *SELECT*FROM DBA_TAB_PRIVS WHERE GRANTABLE = 'YES'*	**YES**	**NO**	**YES**
6.	***PUBLIC* Permissions**	The following queries list the privileges assigned to the *PUBLIC* group: • *SELECT OWNER, TABLE_NAME, GRANTOR, PRIVELEGE FROM DBA_TAB_PRIVS WHERE GRANTEE='PUBLIC' AND GRANTOR <> 'SYSTEM' AND GRANTOR <>'SYS'* • *SELECT GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTEE='PUBLIC'*	**YES**	**NO**	**YES**

(Sources: Huey, 2016; ISACA, 2009)

5.6 Database auditing and monitoring

Database monitoring is one of the controls that can be implemented by management to identify malicious attacks or unauthorised activities in a database (Huey, 2016). Database auditing is typically used to record user activity, and could be used to hold users accountable for specific actions (Huey, 2016). Users (or others such as intruders) may also be deterred from unauthorised activities, considering that they can be held accountable for their activities (Huey, 2016). Furthermore, the investigation of suspicious activity is aided by detailed audit trails, while management and/or auditors can be notified of suspicious or unauthorised activities (Huey, 2016).

International regulations such as the Payment Card Industry Data Security Standard (PCI DSS), the International Convergence of Capital Measurement and Capital Standards: A Revised Framework (Basel II) and the Sarbanes-Oxley Act of 2002 require that access to sensitive data be monitored (Huey, 2016). Therefore, traditional and continuous audit procedures should include verifying that database auditing options are enabled, and that they are protected and monitored for unauthorised changes.

Database monitoring includes database auditing (logging) as well as automated alerts using triggers and stored procedures (Davis *et al*., 2011). Monitoring controls are also extended to capacity management and performance monitoring (Davis *et al*., 2011), which is excluded from the scope of this study, as it does not relate to the integrity or security, but rather to the availability of the database.

Database auditing varies between different products, versions and installations. This is briefly summarised in paragraph 5.6.1, whereafter the audit procedures relevant to database auditing are discussed in paragraphs 5.6.2 and 5.6.3.

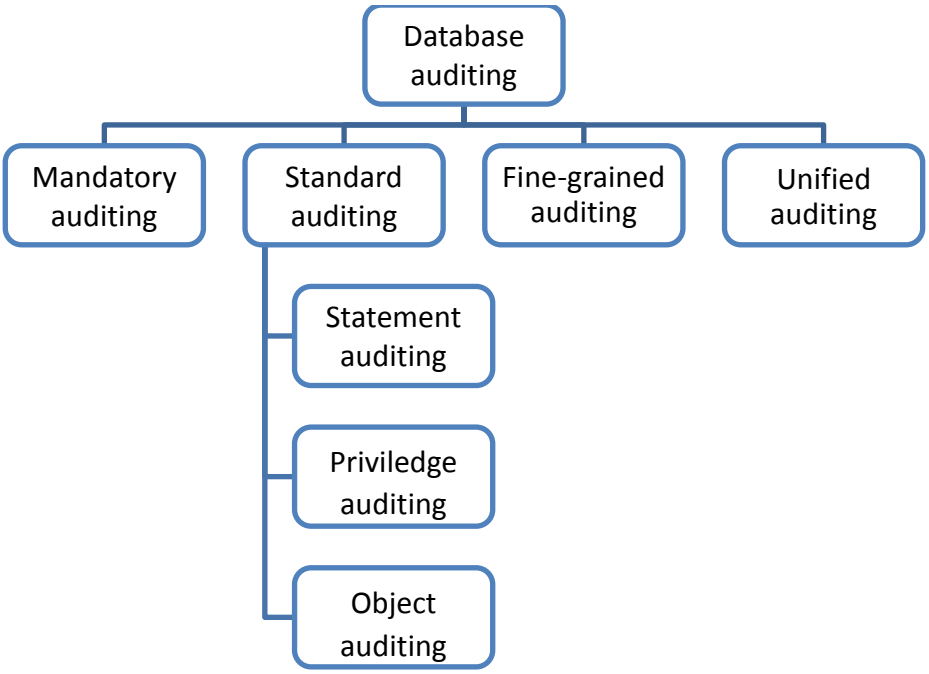
5.6.1 Types of database auditing

Database auditing is the process of recording and monitoring selected database actions, by both database users and non-database users (Dean, 2015). Information such as the event type (e.g. *SELECT TABLE*) is combined with the event context (such as the event time, user name and initiating internet protocol (IP) address) to create an audit trail of sensitive transactions (Chaudhari & Bakal, 2015). For example, an organisation may choose to audit every user attempt to view the contents of a particular table (e.g. credit

card records) or any attempts (both successful and unsuccessful) to access information in the *DBA_USERS* view (ISACA, 2009).

Database auditing is broadly categorised as mandatory, standard, fine-grained and unified (Oracle 12c only) auditing, as shown in Figure 5.3 (Chaudhari & Bakal, 2015). Each type of auditing is described below, while the auditing parameters and storage location of audit logs for each type of auditing are depicted in Figure 5.4.

Figure 5.3 Types of database auditing



(Source: Dean, 2015)

Mandatory auditing

Mandatory auditing, which cannot be disabled, records any actions by users with database administrator privileges *(SYSDBA* and *SYSOPE*R), and will also record when the database is stopped or started (Dean, 2015; Chaudhari & Bakal, 2015). Mandatory auditing has been extended for Oracle 12c, which now records changes to audit-related activities such as creating and altering the audit policies as well as attempts to alter the audit trail table (Miller & Kost, 2016).

Standard auditing

Standard (native) auditing is typically included with most databases and therefore is likely to be inexpensive (Davis *et al*., 2011). Standard auditing is enabled by setting the audit

trail parameter and by configuring auditing for specific statements, privileges and objects. Standard auditing includes statement, privilege and object auditing (Dean, 2015).

- **Statement auditing** enables the broad auditing of SQL statements by type of statements, e.g. *AUDIT TABLE* will log all actions where the *TABLE* statement was used, regardless of which table was accessed (Huey, 2016).
- **Privilege auditing** is more focused than statement auditing and logs only a particular type of action, e.g. *AUDIT CREATE TABLE* will log only those actions that entailed creating a table (Huey, 2016). Newer versions of Oracle Database (i.e. 11g and later) audit selected events by default, while privilege auditing should specifically be configured for older versions (ISACA, 2009).
- **Object auditing** is limited to specific statements on a particular schema object, e.g. *AUDIT SELECT ON EMPLOYEES* will log actions where the table containing employee details was selected (e.g. viewed or copied) (Huey, 2016).

Fine-grained auditing

Oracle Database auditing can be customised on a granular level by following a risk-based approach (ISACA, 2009). This is referred to as fine-grained auditing. Fine-grained auditing enables organisations to create policies that define specific conditions that must take place for the audit to occur (Dean, 2015). Fine-grained auditing can be used to identify suspicious activity such as the following (Huey, 2016):

- Accessing a particular table outside standard office hours;
- Using an IP address from outside the origination's IP range;
- Changing a value in a specific table column (e.g. salaries).

Fine-grained auditing can be combined with event handlers. For example, when suspicious activity is logged as described above, an e-mail alert could be sent to an auditor or database administrator to follow up (Huey, 2016).

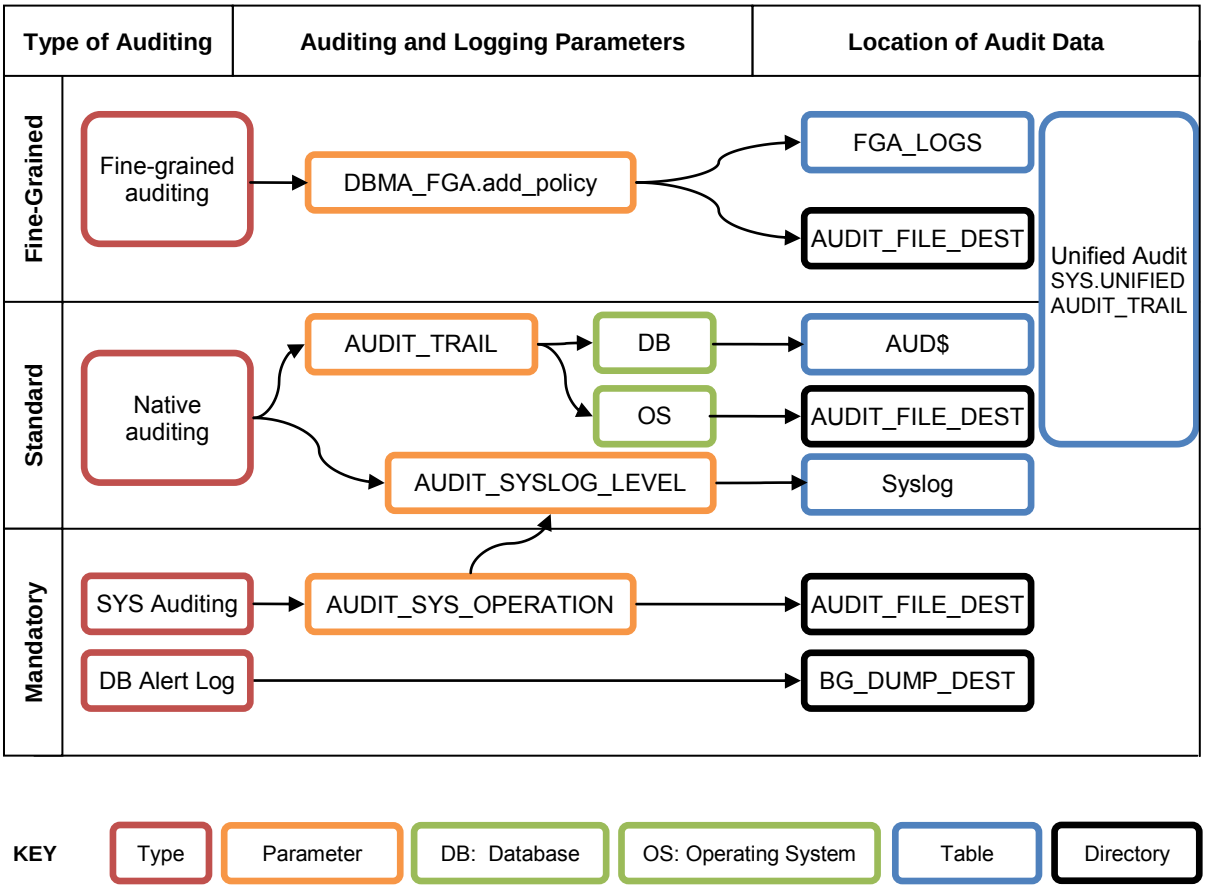
Unified auditing

In addition to the above three general auditing types that are typical of most database management systems, Oracle 12c introduced a new auditing functionality named unified auditing (Dean, 2015). Unified auditing consolidates all auditing configuration and logs in a single location and format (Dean, 2015). To assist organisations in transitioning to

unified auditing, the mixed mode of auditing is enabled by default for Oracle 12c. In the mixed mode, all prior logging and auditing functionalities are available as for prior versions of Oracle Database, in addition to the unified audit functionality (Miller & Kost, 2016). As the mixed mode of auditing is installed by default for Oracle 12c and also addresses the legacy versions of Oracle Database, this mode of auditing was used to develop continuous auditing procedures in this study.

Figure 5.4 depicts the three types of auditing and the location of the audit data for the different auditing and logging parameters. For example, fine grained auditing parameters are configured in the *DBMA_FGA.add_policy* table (Miller & Kost, 2016). When the mode of auditing is selected, audit logs are maintained in both the legacy locations, namely the *FGA_LOGS* database table and the external location specified in *AUDIT_FILE_DEST*, as well as the unified audit trail namely *SYS.UNIFIEDAUDIT_TRAIL* (Miller & Kost, 2016).

Figure 5.4 Oracle unified auditing – mixed mode



(Source: Miller & Kost, 2016)

5.6.2 Enabling database auditing

Risks

Database auditing should be enabled to detect unauthorised activities that may occur on the database (Dean, 2015). The auditing strategy and policies are typically determined by the organisation following a risk assessment for the particular database (ISACA, 2009). The risks related to database auditing include failure to log high-risk events appropriately, including unauthorised access and changes to the audit parameters and inadequate monitoring of high-risk events (ISACA, 2009). In particular, the following should be taken into consideration:

- Database auditing is often disabled, as the enabling thereof will increase storage space requirements, and may adversely affect the performance of the database (ISACA, 2009).
- Although default audit parameters have been extended in Oracle 12c, these default audit parameters may still be disabled by the database administrator (Dean, 2015).
- The privileged *SYS* users are exempt from auditing by default. If this parameter is not enabled, critical activities performed by the *SYS* user are not recorded in the audit trail (Miller & Kost, 2016).
- Unauthorised changes of auditing parameter settings may be processed by privileged users who have the ability to change auditing parameter settings (Larner, 2014). For instance, the Oracle Database *AUDIT SYSTEM* privilege allows the user to change auditing activities such as disabling the creation of audit trails (CIS, 2015).

Controls

Organisations should follow a risk-based approach for enabling auditing on any particular system (ISACA, 2009). The approach to database auditing and monitoring should be documented in a policy document that should address both database changes and high-risk actions related to sensitive data (Miller & Kost, 2016).

As changes to auditing parameters may be abused to conceal unauthorised activities, access to changing auditing parameters should be restricted to privileged users only (Miller & Kost, 2016). These auditing parameter changes should also be logged as part of the audit trail (Miller & Kost, 2016). Privilege auditing is relevant in this scenario and will log all actions where the *AUDIT SYSTEM* statement was used (Dean, 2015).

Database changes introduce the risk of unauthorised changes and include actions such as *ALTER SYSTEM*, data definition language (DDL) statements, using system and object privileges, logon and logoff attempts (successful and failed) and any unsuccessful operations (Miller & Kost, 2016). These actions should be described in database policies and detailed in database security standards (CIS, 2015). Actions that entail changing the database are audited using the mandatory and standard auditing (statement and object) functions described in paragraph 5.6.1 (Miller & Kost, 2016).

Depending on the classification of the data retained in a particular database, auditing should be enabled for any data that is confidential or sensitive. Following a risk assessment in consultation with the business owner, those data objects that present the greatest risk to the organisation should be identified in this manner (ISACA, 2009). Such data objects may include confidential or regulated information such as salaries, credit card numbers, personal information and financial (trading) information (Huey, 2016). The risk assessment requires an in-depth understanding of the data and associated risks and may be an area where object auditing and fine-grained auditing may be most appropriate (Miller & Kost, 2016).

Procedures

Both traditional and continuous auditing procedures will commence with a policy and standards review process to assess the organisation's strategy for enabling database auditing (ISACA, 2009).

As **traditional audit procedures** focus on the manual review of parameter settings of the database, a similar approach is recommended for **continuous auditing procedures**. Continuous auditing will commence in the configuration phase and should extend to the operate and maintain phases to detect any changes to the parameter settings (Cooke, 2014), as described below:

- The first audit procedure is aimed at determining whether auditing is enabled. The *AUDIT_TRAIL* parameter indicates whether auditing is enabled (ISACA, 2009). Refer to Table 5.7 item 1.
- As auditing is not always enabled for default privileged accounts, procedures should include checking whether auditing is enabled for the *SYS*, *SYSOPER, SYSDBA and DBA* accounts (Miller & Kost, 2016). Refer to Table 5.7 item 2.

- As the *AUDIT SYSTEM* privilege enables the alteration of system audit activities, such as disabling the creation of audit trails, this capability should be restricted to privileged administrator accounts only (CIS, 2015). To determine which users can change the auditing parameters, the same tables extracted for user account management (paragraph 5.4.1) are analysed to determine users with this privilege (CIS, 2015). Refer to Table 5.7 item 3.
- To determine which statements, privileges and objects are audited, the auditor should review the database views listed in Table 5.7 items 4 to 6. The relevant database views should be assessed, considering the risks associated with the particular database (ISACA, 2009). Appendix 2 lists the audit queries to confirm whether the minimum auditing requirements recommended by CIS (2015) are activated for statement, privilege and object auditing.
- Audit procedures should extend to the fine-grained auditing parameters that are configured, depending on the risk assessment for the particular database. As the auditing fine-grained auditing parameters are customised for every organisation and database installation (Miller & Kost, 2016), the continuous auditing procedures will also vary for each instance. Therefore, detailed audit procedures for fine-grained auditing are not tabled for the purposes of this study. The procedures for extracting the audit parameters and logs are described in Table 5.7 item 7.
- The auditor should review the audit trails based on the risk assessment for the specific database (ISACA, 2009). The *DBA_AUDIT_TRAIL* lists all audit entries in the *AUD$* table and the *DBA_FGA_AUDIT_TRAIL* lists all audit records related to fine-grained auditing (ISACA, 2009). As the risks and auditing parameters vary between organisations and between each database installation, the continuous auditing procedures will also vary for each instance. Therefore, the audit procedures for analysing audit trails are not detailed in this study.

5.6.3 Protecting the audit trail

Risks

As the audit trail may serve as evidence of unauthorised activities, malicious users may attempt to modify the audit trail entries to conceal their activities (ISACA, 2009). Similarly, privileged users may temporarily deactivate auditing to avoid the logging of their activities (Wright, 2014).

Controls

When auditing for suspicious database activity, the integrity of the audit trail records should be protected using access controls to guarantee the accuracy and completeness of the auditing information (Larner, 2014). Any successful and failed attempts to change audit configuration settings and content of audit trails should also be logged (CIS, 2015). However, system administrators can change audit settings and modify audit trails even though the activities of such accounts are also logged (Wright, 2014). This inherent weakness of Oracle Database was addressed in Oracle 12c by the introduction of Oracle Vault as well as a separate role (*AUDIT_ADMIN*) to administer audit policies, and the use of *AUDIT* and *NOAUDIT* statements (Miller & Kost, 2016).

Procedures

Both **traditional** and **continuous auditing procedures** will commence with a policy and standards review process to assess the organisation's strategy for protecting the audit trail. As the procedures will vary depending on the auditing options implemented by the organisation, the following should be tested as a minimum:

- Determine whether access to the audit trail tables (e.g. *SYS.AUD$* and *SYS.FGA_LOG$)* is restricted to trusted users only. Similarly, confirm that no user have access to the table containing password re-set history (CIS, 2015). Unauthorised users may manipulate this audit table to conceal their attempts to compromise user account passwords (CIS, 2015). Refer to Table 5.7 item 8.
- Determine whether all attempts to access or alter the audit trail are logged. Both successful and failed attempts could indicate that the system is under attack (Huey, 2014). Refer to Table 5.7 item 9.
- Ensure that dictionary accessibility is restricted to trusted users only, similar to the audit trail (CIS, 2015). In this manner, only the *SYSDBA* account is able to use data manipulation language (DML) actions to edit the audit trail (Huey, 2014). Refer to Table 5.7 item 10.
- Detect any interruptions of the audit trail (audit bypass) by comparing the system start-up times to any periods where no activities were logged in the audit trail (Wright, 2014). Refer to Table 5.7 item 11.

All of the above procedures relating to the protection of the audit trail will be performed during the configuration phase to determine the baseline standard for auditing. Any

changes will be detected by repeating the test during the maintain phase. Only the interruption of the audit trail would be relevant in the operate phase when the database is stopped or started.

5.6.4 Stored procedures database triggers

Stored procedures are programs written in Oracle's extension to the SQL programming language, PL/SQL. These programs are used to package defined business transactions and to perform controlled operations on the database using conditional statements (IF-THEN-ELSE) (ISACA, 2009). Stored procedures can be invoked by any user or program with the *EXECUTE* privilege (ISACA, 2009).

Database triggers are similar to stored procedures. The primary difference is that triggers are activated when certain conditions are met, instead of being invoked by a user (ISACA, 2009). Database administrators can actively monitor the database using triggers. For example, an automated e-mail notification could be created for any unsuccessful attempt to access a table with sensitive information (ISACA, 2009).

While Oracle continues support for triggers in later versions (11g and onwards), the use of triggers for internal auditing purposes are limited (ISACA, 2009). In particular, triggers cannot be created for *SELECT* statements (ISACA, 2009). Furthermore, triggers may have a performance impact on the system and should only be used for non-transactional tables (ISACA, 2009). Finally, database triggers are not protected from certain privileged users such as the *SYS* account, which has the ability to disable or change triggers and custom audit trail tables (Larner, 2014).

To overcome these limitations, Oracle have implemented fine-grained auditing that supports a more granular audit trail and can also actively alert administrators of policy violations (ISACA, 2009). Bearing in mind the above control weaknesses and the better alternative offered through the use of fine-grained auditing, database triggers are not considered a reliable method to obtain audit evidence, and are therefore excluded from the scope of this study.

Table 5.7 Continuous audit procedures: Database monitoring and auditing

Risk/Control Area	SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
		Configure	Operate	Maintain
1. Determine whether auditing is enabled	Extract the setting for the *AUDIT_TRAIL* parameter in the *init<SID>.ora* file. Alternative query script: *SELECT* FROM V$PARAMETER WHERE NAME = 'audit_trail'* Auditing is enabled if the result is: • *DB, EXTENDED* or *TRUE* (stored in *AUD$* table), • *OS* (stored in operating system file) or • *XML, EXTENDED*	**YES**	**NO**	**YES**
2. Determine whether auditing is enabled for privileged accounts (*SYS*, *SYSDBA*, *SYSOPER*, *DBA*)	Extract the setting of *AUDIT_SYS_OPERATIONS*: *SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME) = 'AUDIT_SYS_OPERATIONS'* Auditing of these accounts is enabled if set to TRUE	**YES**	**YES**	**YES**
3. Determine whether users can change the auditing parameters	Extract users with the *AUDIT SYSTEM* privilege which allows changing the auditing activities. *SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS WHERE PRIVILEGE='AUDIT SYSTEM' AND GRANTEE NOT IN ('DBA', 'DATAPUMP_IMP_FULL_DATABASE', 'IMP_FULL_DATABASE', 'SYS','AUDIT_ADMIN')*	**YES**	**NO**	**YES**

Risk/Control Area	SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
		Configure	Operate	Maintain
4. Review privilege level auditing	The following query result indicates whether privilege level auditing is enabled: *SELECT*FROM DBA_PRIV_AUDIT_OPTS* Refer to Appendix 2 for a list of recommended queries.	**YES**	**NO**	**YES**
5. Review statement- level auditing	The following query result indicates whether statement level auditing is enabled: *SELECT*FROM DBA_STMT_AUDIT_OPTS* Refer to Appendix 2 for a list of recommended statement level queries.	**YES**	**NO**	**YES**
6. Review object-level auditing	The following query result indicates whether object level auditing is enabled: *SELECT*FROM DBA_OBJ_AUDIT_OPTS* Refer to Appendix 2 for a list of recommended object level queries.	**YES**	**NO**	**YES**
7. Fine-grained auditing	Fine-grained auditing parameters are stored in *DBA_AUDIT_POLICIES* and logs are retained in the *FGA_LOG$* table or can be extracted using the *DBA_FGA_AUDIT_TRAIL.*	**YES**	**NO**	**YES**

Risk/Control Area	SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
		Configure	Operate	Maintain
8. Prevent access to audit trail tables	Where the audit trail parameter is set to *DB* or *DB_EXTENDED* (refer to Table 5.7 item 2), access to the *AUD$* table should be tested. Similarly, access to the *USER_HISTORY$* table (containing the history of password changes) should be tested. *SELECT GRANTEE, PRIVILEGE FROM DBA_TAB_PRIVS WHERE TABLE_NAME="AUD$"* *SELECT GRANTEE, PRIVILEGE FROM DBA_TAB_PRIVS WHERE TABLE_NAME="USER_HISTORY$"*	**YES**	**NO**	**YES**
9. Audit all attempts to access or alter the audit trail (AUD$ table)	This test is only applicable where the audit trail (*AUD$*) parameter is set to *DB* or *TRUE* (refer to Table 5.7 item 2) *SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OBJECT_NAME ="AUD$"*	**YES**	**NO**	**YES**
10. Ensure dictionary accessibility is restricted	Determine whether dictionary accessibility initialisation parameters are set to FALSE. *SELECT UPPER(VALUE) FROM V$PARAMETER WHERE UPPER(NAME)='O7_DICTIONARY_ ACCESSIBILITY'* Ensure that the VALUE is set to FALSE.	**YES**	**NO**	**YES**

Risk/Control Area	SQL*Plus query statement or DBA view extracted to obtain Oracle configuration	Procedure relevant to particular product lifecycle phase?		
		Configure	Operate	Maintain
11. Detect interruption of the audit trail (audit bypass)	Any interruptions in the SYS audit trail can be detected by matching "quiet times" in the audit trail to the database restart times collected remotely using this SQL statement: *SELECT STARTUP_TIME FROM DBA_HIST_ PDB_INSTANCE;STARTUP_TIME*	**YES**	**YES**	**YES**

(Sources: Huey, 2016; Miller & Kost, 2016; Dean, 2015; CIS, 2015, Wright, 2014)

5.7 Conclusion

Using the audit planning framework developed in this study (refer to Figure 4.10), continuous auditing procedures were developed at an operational level for four of the identified control categories for database management systems (depicted in Figure 5.1). Using Oracle Database as example, it was demonstrated that continuous auditing procedures relating to configuration controls can be evaluated continuously against a baseline standard to identify configuration changes and risk indicators.

The proposed continuous auditing procedures are relevant in the configuration phase, when determining the baseline standard for configuration controls. Thereafter, the scripted extracts are repeated periodically (or continuously) during the operate and maintain phases of the product's lifecycle.

Although the continuous auditing procedures were developed using Oracle Database as example, the same methodology could also be used to develop procedures for other commercially available database management systems.

CHAPTER 6. CONCLUSION

King III has emphasised the responsibility of audit committees to ensure that IT risks are adequately governed through risk management, monitoring and assurance processes (IODSA, 2009). While the use of technology to improve audit coverage and efficiency is recommended by King III, no specific guidance is provided (IODSA, 2009).

In response to this recommendation by King III, the primary objective of this study was to develop a generic audit planning framework at a strategic and operational level to assist internal auditors in implementing continuous auditing for automated IT controls.

At an operational level, the planning framework is based on the identification of the applicable IT architecture components, by identifying all possible access points of an IT access path (Boshoff, 1990). Individual components are analysed to determine the baseline standard of controls (IIA, 2015) and the relevant lifecycle phases (Boshoff, 2014). When measured continuously against this baseline standard, any subsequent changes in configuration settings can be highlighted for further investigation by internal audit (IIA, 2015). The generic audit planning framework developed in this study can also be used by internal auditors to implement continuous auditing procedures for any IT architecture component on an operational level.

The secondary objective of this study was to apply the above framework at an operational level for one of the IT access path components, namely database management systems. In particular, continuous auditing procedures were developed to provide assurance on the validity, integrity and confidentiality of database management systems, using Oracle Database as example. The procedures were designed to be conducted by using generalised audit software which is widely used by internal audit functions. These automated audit procedures were developed for the different lifecycle phases for four typical control categories for database management systems, namely:

- Database vulnerabilities;
- Account and password management;
- Permissions management; and
- Database auditing and monitoring.

These procedures, which were developed specifically for Oracle Database, can be adapted for most commercial database management systems by any organisation that adopts this audit methodology, considering the organisational and business process risks.

Since this study focused on detailed continuous auditing procedures for only one component of an IT access path, namely database management systems, the remaining IT access paths components remain available for further research. Similarly, advanced database security measures, which were excluded from the scope of this review, may also be suited to further research on continuous auditing.

By implementing continuous auditing procedures for IT architecture components, internal auditors are enabled to report on control failures within a shorter timeframe, potentially instantaneously, potentially resulting in real-time assurance (Soileau *et al*., 2015). The efficiency of such audits is also improved through automation of processes, while audit coverage and effectiveness may also increase (IIA, 2015). Considering that the most valuable information assets of organisations are retained in databases and in view of the increase in data breaches of high-profile organisations (McAfee, 2015), the implementation of continuous auditing for database management systems should be of high priority for internal audit functions.

Continuous auditing is a potential solution to address the perceived failure of internal audit functions on meeting stakeholder expectations and delivering on future demands (Deloitte, 2016). Chief audit executives observed that internal audit functions currently do not have the desired impact and influence within the organisation, partially because of skills shortages relating to analytics, IT and communications (Deloitte, 2016). Similarly, internal audit's stakeholders expect forward-looking reports that provide insights regarding risk, strategic planning, IT and business performance (Deloitte, 2016).

By investing in continuous auditing methodologies as demonstrated in this study, internal audit functions will be able to stay relevant in the modern business environment.

APPENDIX 1 – ACCESS PATH COMPONENTS

The **application program code** includes the sets of computer programs, control files, tables and user interfaces which provide functionality for specific business operations such as accounting, payroll and procurement. The application layer is the component which is visible to and accessed by the end-user (Davis *et al.,* 2011).

Database management systems enable the storage, modification, and extraction of data (IIA, 2008). This tool organises and provides access to the data to be used by the application, such as Oracle Database, Microsoft SQL Server, DB2 (Davis *et al*., 2011).

Operating systems perform a computer's basic tasks, such as managing operator input, managing internal computer memory and providing disk drive, display and peripheral device functions (e.g. Windows, Linux, UNIX and iOS) (IIA, 2008).

Networks link computers and the system's users and enable the communication of network components, whether across a wire, fibre-optic cable or wireless network (IIA, 2013b). The physical network layer includes devices such as firewalls, switches, routers and wiring. Networks also include the programs which control the routing of data packets (Davis *et al.,* 2011).

Data centre facilities encompass the physical building and data centre housing the computer equipment on which the system resides (Davis *et al*., 2011).

APPENDIX 2 – DATABASE AUDITING PARAMETERS

Statement Auditing

1. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'USER'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

2. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'ALTER USER'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

3. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'DROP USER'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

4. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'ROLE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

5. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'SYSTEM GRANT'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

6. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'PROFILE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

7. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'ALTER PROFILE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

8. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'DROP PROFILE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

9. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'DATABASE LINK'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

10. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'PUBLIC DATABASE LINK'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

11. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'PUBLIC SYNONYM'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

12. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'SYNONYM'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

13. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'GRANT DIRECTORY'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

14. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'SELECT ANY DICTIONARY'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

15. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'DROP ANY PROCEDURE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

16. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'TRIGGER'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

17. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'CREATE SESSION'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

18. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'PROCEDURE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

19. SELECT AUDIT_OPTION, SUCCESS, FAILURE FROM DBA_STMT_AUDIT_OPTS WHERE AUDIT_OPTION=**'ALTER SYSTEM'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

Privilege Auditing

20. SELECT PRIVILEGE, SUCCESS, FAILURE FROM DBA_PRIV_AUDIT_OPTS WHERE PRIVILEGE=**'GRANT ANY OBJECT PRIVILEGE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

21. SELECT PRIVILEGE, SUCCESS, FAILURE FROM DBA_PRIV_AUDIT_OPTS WHERE PRIVILEGE=**'GRANT ANY PRIVILEGE'** AND USER_NAME IS NULL AND PROXY_NAME IS NULL AND SUCCESS = 'BY ACCESS' AND FAILURE = 'BY ACCESS'

Object Auditing

22. SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OBJECT_NAME=**'AUD$'** AND ALT='A/A' AND AUD='A/A' AND COM='A/A' AND DEL='A/A' AND GRA='A/A' AND IND='A/A' AND INS='A/A' AND LOC='A/A' AND REN='A/A' AND SEL='A/A' AND UPD='A/A' AND FBK='A/A'

(Source: CIS, 2015)

REFERENCES

AICPA (American Institute of Certified Public Accountants (ed.)). 2015. *Audit analytics and continuous auditing: Looking toward the future.* New York: American Institute of Certified Public Accountants.

Alles, M. G., Brennan, G., Kogan, A. & Vasarhelyi, M. A. 2006. Implementation of a continuous auditing system at Siemens. *International Journal of Accounting Information Systems,* 7(2): 137-161.

Alles, M. G., Kogan, A. & Vasarhelyi, M. A. 2011. Collaborative design research: Lessons from continuous auditing. *International Journal of Accounting Information Systems,* 14(2): 104-112.

AuditNet. 2012. *AuditNet 2012 survey report on data analysis audit software.* [Online] Available at: www.auditnet.org/system/2012 DataAnalyticsAuditSoftware Survey.pdf [Accessed: 18 July 2015].

AuditNet. 2015. *2015 Data analysis audit software survey.* [Online] Available at: https://www.surveymonkey.com/results/SM-WYX6VMW2/ [Accessed: 10 October 2015].

Boshoff, W. 2014. *Master's in Commerce (Computer Auditing).* Unpublished lecture notes. Stellenbosch: Stellenbosch University.

Boshoff, W. H. 1990. *A path context model for computer security phenomena in potentially non-secure environments.* Unpublished doctoral dissertation. Johannesburg: Rand Afrikaans University.

Bumgarner, B. & Vasarhelyi, M. A. 2015. Continuous auditing – A new view, in AICPA (ed.). *Audit analytics and continuous audit: Looking toward the future.* New York: American Institute of Certified Public Accountants: 3-52.

Byrnes, P. E. Ames, B., Vasarhelyi, M. A., Pawlicki, A., & McQuilken, D. 2015a. The current state of continuous auditing and continuous monitoring, in AICPA (ed.). *Audit analytics and continuous audit: Looking toward the future.* New York: American Institute of Public Accountants: 53-70.

Byrnes, P. E., Al-Awahdi, A., Gullvist, B., Brown-Liburd, H., Teeter, R., Warren, J. D., & Vasarhelyi, M. 2015b. Evolution of auditing: From the traditional approach to the future audit, in AICPA (ed.). *Audit analytics and continuous audit: Looking toward the future.* New York: American Institute of Public Accountants: 71-85.

Byrnes, P.E., Brennan, G., Vasarhelyi, M. A. & Moon, D. 2015c. Managing risk and the audit process in a world of instantaneous change, in AICPA (ed.). *Audit analytics and continuous audit: Looking toward the future.* New York: American Institute of Public Accountants, Inc.: 129-143.

Cangemi, M. P. 2015. *Staying a step ahead – Internal audit's use of technology.* [Online] Available at: http://contentz.mkt5790.com/lp/2842/191428/2015-1403_CBOK_Staying AStepAhead.pdf [Accessed: 3 August 2016].

Cangemi, M. P. 2016. *Views on internal audit, internal controls, and internal audit's use of technology.* [Online] Available at: https://www.knowledgeleader.com/KnowledgeLeader/ Resources.nsf/Description/ViewsonInternalAuditInternalControls/$FILE/Views%20on%20I nternal%20Audit%20Internal%20Controls.pdf [Accessed: 3 August 2016].

Cascarino, R. E. 2012. *Auditor's guide to IT auditing.* 2nd edition. New Jersey: John Wiley & Sons.

CEB (Corporate Executive Board). 2015. *2015 Audit department challenges and priorities.* [Online] Available at: https://audit.executiveboard.com/Members/ ResearchAndTools/ Abstract.aspx?cid=101269954&fs=1&q=audit%20strategy&program=&ds=1&acws=WS_R RES_RS [Accessed: 6 June 2015].

Chan, D. & Vasarhelyi, M. 2011. Innovation and practice of continuous auditing. *International Journal of Accounting Information Systems,* 12: 152-160.

Chaudhari, R. & Bakal, J. 2015. Overview of database auditing for Oracle database. *International Journal of Application or Innovation in Engineering & Management,* 4(7): 189-196.

Chiu, V., Liu, Q. & Vasarhelyi, M. 2014. The development and intellectual structure of continuous auditing research. *Journal of Accounting Literature,* 33: 37-57.

CICA & AICPA (Canadian Institute of Chartered Accountants & American Institute of Certified Public Accountants. 1999. *Continuous auditing: Research report,* Toronto: Canadian Institute of Chartered Accountants.

CIS (Center for Internet Security). 2015. *The security configuration benchmark for Oracle database server 12c.* [Online] Available at: https://security.uri.edu/uploads/CIS_Oracle_Database_12c_Benchmark_v1.0.0.pdf [Accessed: 7 April 2016].

Cooke, I. 2014. Auditing Oracle databases using CAATs. *ISACA Journal,* 2: 25-28.

Cooke, I. 2015. Auditing SQL Server databases using CAATs. *ISACA Journal,* 2: 38-43.

Corderre, D. 2010. *Internal audit practice guide: Continuous auditing.* [Online] Available at: www.centerforcontinuousmonitoring.org/wp-wpcontent/uploads/ 2011/01/Internal-Audit-Practice-Guide-Continuous-Auditing-Dave-Corderre.pdf [Accessed: 26 July 2015].

Davis, C., Schiller, M. & Wheeler, K. 2011. *IT auditing: Using controls to protect information assets.* 2nd edition. New York: MacGraw-Hill.

Dean, M. 2015. *All about Oracle auditing – Updated for 12c.* [Online] Available at: www.dbspecialists.com/files/presentations/OracleAuditing-Whitepaper.pdf [Accessed: 26 June 2016].

De Kroon, N. & Karp, B. 2013. *An auditor's guide to data analytics.* Raleigh, ISACA.

Deloitte. 2016. *2016 Global chief audit executive survey.* [Online] Available at: www2.deloitte.com/global/en/pages/audit/solutions/global-chief-audit-executive-survey.html [Accessed: 7 August 2016].

Feinberg, D., Adrian, M., Heudecker, N., Ronthal, A. M. & Palanca, T. 2015. *Magic quadrant for operational database management systems.* [Online] Available at: https://www.gartner.com/doc/reprints?id=1-2PMFPEN&ct =151013&st=sb [Accessed 27 August 2016].

Finnigan, P. 2016. *Oracle Default Password List.* [Online] Available at: http://www.petefinnigan.com/default/default_password_list.htm [Accessed: 1 May 2016].

Flowerday, S., Blundell, A. W. & Von Solms, R. 2006. Continuous auditing technologies and models: A discussion. *Computers and Security,* 25(5): 325-331.

Gibbs, N., Jain, D., Joshi, A., Muddamsetti, S. & Singh, S. 2010. *A new auditor's guide to planning, performing, and presenting IT audits.* Altamonte Springs: Institute of Internal Auditors Research Foundation.

Gonzalez, G. C., Sharma, P. S. & Galletta, D. F. 2012. The antecedents of the use of continuous auditing in the internal audit context. *International Journal of Accounting Information Systems,* 13: 248-262.

Goosen, R. 2012. *The development of an integrated framework in order to implment information technology governance principles at a strategic and operational level for medium-to-large sized South African Businesses.* Unpublished Master's thesis. Stellenbosch: Stellenbosch University.

Goosen, R. & Rudman, R. 2014. Practical implementation of IT governance. *Auditing SA,* Summer 2013/2014: 60.

Groomer, S. & Murthy, U. 1989. Continuous auditing of database applications: An embedded audit module approach. *Journal of Information Systems,* 3(2): 53-69.

Hargenrader, B. 2015. Information security continuous monitoring. *ISACA Journal,* 1: 48-52.

Hoehl, M. 2013. *Framework for building a comprehensive enterprise security patch management program.* [Online] Available at: https://www.sans.org/reading-room/whitepapers/threats/framework-building-comprehensive-enterprise-security-patch-management-program-34450 [Accessed: 30 April 2016].

Huey, P. 2014. *Oracle database security guide 11g Release 1 (11.1).* [Online] Available at: https://docs.oracle.com/cd/B28359_01/network.111/b28531/title.htm [Accessed: 11 September 2016].

Huey, P. 2016. *Oracle database security guide 12c Release 1 (12.1).* [Online] Available at: https://docs.oracle.com/database/121/DBSEG/title.htm [Accessed: 5 June 2016].

IIA (Institute of Internal Auditors). 2005. *Global technology audit guide 3: Continuous auditing: Implications for assurance, monitoring and risk assessment,* Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors). 2008. *Global technology audit guide 11: Developing the IT audit plan.* Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors). 2011. *Global technology audit guide 16: Data analysis technologies,* Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors). 2012. *Global techonology audit guide 2: Change and patch management controls: Critical for organisational success.* 2nd edition. Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors). 2013a. *International professional practices framework.* Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors). 2013b. *Global technology audit guide 4: Management of IT auditing.* Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors). 2015. *Global technology audit guide 3: Coordinating continuous auditing and monitoring to provide continuous assurance.* 2nd edition. Altamonte Springs: Institute of Internal Auditors.

IIA (Institute of Internal Auditors) Research Foundation. 2015. *The global internal audit common body of knowledge.* [Online] Available at: https://global.theiia.org/iiarf/pages/common-body-of-knowledge-cbok.aspx [Accessed: 3 August 2016].

IODSA (Institute of Directors in Southern Africa). 2009. *King report on governance for South Africa 2009,* Institute of Directors in Southern Africa.

ISACA. 2009. *Security, audit and control features: Oracle database.* 3rd edition. Rolling Meadows: ISACA.

ISACA. 2010. *IT Audit and assurance guidelines: G42 continuous assurance,* Rolling Meadows: ISACA.

ISACA. 2011. *Data analytics – A practical approach,* Rolling Meadows: ISACA.

ISACA. 2014. *Professional Practice Framework for IS Audit/Assurance*. 3rd ed. Rolling Meadows: ISACA.

ISACA. 2016. *Glossary.* [Online] Available at: www.isaca.org/Pages/ Glossary.aspx?tid=1247&char=C [Accessed: 30 April 2016].

ISACA Standards Board. 2002. Continuous auditing: Is it fantasy or reality? *ISACA Journal,* 5: 43-46.

KPMG. 2009. *IT audit perspective on continuous auditing/continuous monitoring.* [Online] Available at: https://www.kpmg.com/PL/pl/services/Advisory/Ryzyko-i-zgodnosc/Ciagly-Audyt-i-Ciagly-Monitoring-CACM/Documents/IT-Audit-Perspective-on-Continuous-Auditing-Continuous-Monitoring-secured.pdf. [Accessed: 29 October 2016].

KPMG. 2013. *Transforming internal audit: A maturity model from data analytics to continuous assurance.* [Online] Available at: https://www.kpmg.com/BE/en/IssuesAndInsights/ArticlesPublications/Pages/Transforming-Internal-audit.aspx [Accessed: 28 October 2016].

KPMG. 2015. *Leveraging data analytics and continuous auditing processes for improved audit planning, effctiveness and efficiency.* [Online] Available at: www.kpmg.com/za/en/issuesandinsights/articlespublications/risk-compliance/pages/leveraging-data-analytics-continuous-auditing/aspx [Accessed: 29 October 2016].

Larner, C. 2014. *Auditing the DBA in Oracle applications: A guide for compliance and audit managers.* [Online] Available at: http://www.absolute-tech.com/wp-content/uploads/2014/04/Absolute-WP- Auditing-the- DBA-20081.pdf [Accessed: 29 October 2016].

Le Roux, J. & Wallis, N. 2014. Getting 'fit for fraud and abuse'. *Auditing SA,* Summer 2013/2014: 57-60.

Lindros, K. & Tittel, E. 2014. *How to choose the best vulnerability scanning tool for your business.* [Online] Available at: http://www.cio.com/article/2683235/security0/ how-to-choose-the-best-vulnerability-scanning-tool-for-your-business.html [Accessed: 3 August 2016].

McAfee. 2015. *Grand Theft Data - Data exfiltration study: Actors, tactics, and detection.* [Online] Available at: www.mcafee.com/cn/resources/reports/rp-data-exfiltration.pdf [Accessed: 25 October 2016].

Microsoft. 2013. *New cybersecurity report details risk of running unsupported software.* [Online] Available at: http://blogs.microsoft.com/on-the-issues/2013/10/29/new-cybersecurity- report-details-risk-of-running-unsupported-software/ [Accessed: 23 April 2016].

Microsoft. 2016. *How to determine the version, edition and update level of SQL Server and its components.* [Online] Available at: https://support.microsoft.com/en-za/kb/321185#/en-za/kb/321185 [Accessed: 24 April 2016].

Miller, M. & Kost, S. 2016. *Oracle 12c Unified auditing.* [Online] Available at: www.integrigy.com/security-resources/oracle-12c-unified-auditing [Accessed: 1 July 2016].

Oracle. 2016. *Lifetime support policy, coverage for Oracle technology products.* [Online] Available at: http://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf [Accessed: 24 April 2016].

Protiviti. 2015a. *2015 Internal audit capabilities and needs survey: Assessing the top priorities for internal audit functions.* [Online] Available at: http://www.protiviti.com/en-US/Pages/IA-Capabilities-and-Needs-Survey.aspx [Accessed: 29 June 2015].

Protiviti. 2015b. *Changing trends in internal audit and advanced analytics.* [Online] Available at: https://www.knowledgeleader.com/KnowledgeLeader/Resources.nsf/Description/2015InternalAuditCapabilitiesandNeedsSurveyProtiviti/$FILE/2015-Internal-Audit-Capabilities-and-Needs-Survey-Protiviti.pdf [Accessed: 29 October 2016].

PwC (PricewaterhouseCoopers). 2013. *2013 State of the internal audit profession.* E-mail to Driekie van Dyk [Online], 29 July 2015. Available e-mail: adiba.khan@us.pwc.com.

PwC (PricewaterhouseCoopers). 2015. *2015 state of the internal audit profession.* [Online] Available at: http://www.pwc.com/us/en/risk-assurance/internal-audit-transformation-study/assets/pwc-state-of-the-internal-audit-profession-2015.pdf [Accessed: 29 October 2016].

Rahman, M. M. 2014. Auditing Oracle database. *ISACA,* 6: 44-50.

Rich, B. 2016. *Oracle database reference, 12c Release 1 (12.1).* [Online] Available at: https://docs.oracle.com/database/121/REFRN/title.htm [Accessed: 16 June 2016].

Rochford, O. & Akshay, L. 2015. *Market guide for vulnerability assessment.* [Online] Available at: https://www.gartner.com/doc/reprints?id=1-32T4TQ6&ct=160404&st=sb [Accessed: 3 August 2016].

Roth, F. 2012. *A look at continuous audting's evolving landscape.* [Online] Available at: https://www.knowledgeleader.com/KnowledgeLeader/Content.nsf/Web+Content/HIALookatContinuousAuditingsEvolvingLandscape!OpenDocument [Accessed: 29 October 2016].

Schultz, M. 2014. Audit-focused mining - New ideas on integrating process mining and internal control. *ISACA Journal,* 3: 45-50.

Soileau, J., Soileau, L. & Sumners, G. 2015. The evolution of analytics and internal audit. *EDPACS: The EDP Audit, Control and Security Newsletter,* 51(2): 10-17.

Teeter, R. A. 2014. *Essays on the enhanced audit.* Unpublished doctoral dissertation. Newark: State University of New Jersey.

Tysiac, K. 2015. *Driving faster decisions.* [Online] Available at: www.journalofaccountancy .com/issues/2015/apr/data-driven-auditing.html [Accessed: 26 July 2015].

Vasarhelyi, M. 1983. A framework for audit automation: Online technology and the audit process. *The Accounting Forum,* January.

Vasarhelyi, M. 1984. Automation and changes in the audit process (practice note). *Auditing: A Journal of Practice and Theory,* 4(1): 100-106.

Vasarhelyi, M. A. & Halper, F. B. 1991. The continuous audit of online systems. *Auditing: A Journal of Practice and Theory,* 10(1): 110-125.

Vasarhelyi, M., Alles, M. & Williams, K. 2010. *Continuous assuarance for the now economy.* Sydney: Institute of Chartered Accountants in Australia.

Warren, J. D. & Parker, X. L. 2003. *Continuous auditing: Potential for internal auditors.* Altamonte Springs: The Institute of Internal Auditors Research Foundation.

Whitehouse, T. 2012. *Putting continuous auditing to use Can yield efficiency and value.* [Online] Available at: http://www.reliantaudit.com/putting-continuous-auditing-to-use-can-yield-efficiency-and-value/ [Accessed: 29 October 2016].

Wright, P. 2014. *Protecting Oracle database 12c.* New York: Apress.